



TITLE FILE COPY

AD-A208 118

(4)

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

VLSI PUBLICATIONS

VLSI Memo No. 89-512
April 1989**Optimal and Near-Optimal Results on Book Embeddings**

Seth Martin Malitz

DTIC
ELECTE
MAY 24 1989
S D**Abstract**

A book embedding of a graph consists of a linear ordering of the vertices along the spine of a book and an assignment of edges to pages so that edges on the same page do not intersect. The minimum number of pages in which a graph can be embedded is its pagenumber. The pagenumber of a class of graphs is the minimum number of pages in which all the members of the class can be embedded, as a function of graph size. In this thesis, we prove the following results, all of which substantially improve previously known bounds.

- (1) The class of E -edge graphs has pagenumber $O(\sqrt{E})$. This result is tight since the complete graph on n vertices has $\Theta(n^2)$ edges and pagenumber $\Theta(n)$. A Las Vegas algorithm to embed any E -edge graph in $O(\sqrt{E})$ pages is also given.
- (2) The class of genus g graphs has pagenumber $O(\sqrt{g})$. This verifies a conjecture due to Heath and Istrail. The result is tight since the complete graph on n vertices has genus $\Theta(n^2)$ and pagenumber $\Theta(n)$. We give a Las Vegas algorithm to produce an $O(\sqrt{g})$ -page embedding when both the graph and its surface embedding are provided as input.
- (3) The $n \times n$ Mesh of Cliques has pagenumber $O(n^{3/2})$. The $n \times n$ Mesh of Cliques is the graph whose vertex set is $\{1, 2, \dots, n\} \times \{1, 2, \dots, n\}$ and whose edges connect each row $\{i\} \times \{1, 2, \dots, n\}$ into an n -vertex clique and each column $\{1, 2, \dots, n\} \times \{i\}$ into an n -vertex clique. The Mesh of Cliques was mentioned by Chung, Leighton, and Rosenberg as a particularly nice example of a regular graph with unknown pagenumber.
- (4) Most n -vertex d -regular graphs have pagenumber $\Omega(\sqrt{dn^{1/2-1/d}})$. This bound is tight when $d > \log n$. The crux of the argument is a lemma that shows most d -regular graphs are "nearly" complete when suitably large clusters of vertices are identified. We suspect this lemma may have application to a variety of other graph embedding problems. For instance, we use it to derive an area lower bound for a multilayer VLSI grid model introduced by Aggarwal, Klawe, and Shor.

DISTRIBUTION STATEMENT A

Approved for public release

Distribution Unlimited

Acknowledgements

Submitted to the Department of Mathematics in partial fulfillment of the requirements for the degree of Doctor of Philosophy at MIT, September 1988. This research was supported in part by NSF grants MIP 83-52349 and CCR86-02062, Army Fellowship BAAL03-86-G-0015, and Air Force contract OSR-86-0078, and the Defense Advanced Research Projects Agency under contract number N00014-87-K-0825.

Author Information

Malitz, *current address*: Computer and Information Sciences Department, University of Massachusetts, Amherst, MA 01003. (413) 545-0111.

Copyright© 1989 MIT. Memos in this series are for use inside MIT and are not considered to be published merely by virtue of appearing in this series. This copy is for private circulation only and may not be further copied or distributed, except for government purposes, if the paper acknowledges U. S. Government sponsorship. References to this work should be either to the published version, if any, or in the form "private communication." For information about the ideas expressed herein, contact the author directly. For information about this series, contact Microsystems Research Center, Room 39-321, MIT, Cambridge, MA 02139; (617) 253-8138.

OPTIMAL AND NEAR-OPTIMAL RESULTS ON BOOK EMBEDDINGS

by

SETH MARTIN MALITZ
B.A., Mathematics
University of Colorado, Boulder
(1983)

Submitted to the
Department of Mathematics
in Partial Fulfillment of the Requirements
for the Degree of
DOCTOR OF PHILOSOPHY
at the
MASSACHUSETTES INSTITUTE OF TECHNOLOGY
September 1988
© 1988 Massachusetts Institute of Technology

Signature of Author

Seth Martin Malitz

Department of Mathematics
September 15, 1988

Certified by

[Signature]

Professor Tom Leighton
Thesis Supervisor

Accepted by

[Signature]

Professor Daniel Kleitman, Chairman
Appointed Mathematics Committee

Accepted by

Sigurdur Helgason

Professor Sigurdur Helgason, Chairman
Departmental Graduate Committee
Department of Mathematics

OPTIMAL AND NEAR-OPTIMAL RESULTS ON BOOK EMBEDDINGS

by

SETH MARTIN MALITZ

Submitted to the
Department of Mathematics
in Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

Abstract:

A book embedding of a graph consists of a linear ordering of the vertices along the spine of a book and an assignment of edges to pages so that edges on the same page do not intersect. The minimum number of pages in which a graph can be embedded is its pagenumber. The pagenumber of a class of graphs is the minimum number of pages in which all the members of the class can be embedded, as a function of graph size. In this thesis, we prove the following results, all of which substantially improve previously known bounds.

- (1) The class of E -edge graphs has pagenumber $O(\sqrt{E})$. This result is tight since the complete graph on n vertices has $\Theta(n^2)$ edges and pagenumber $\Theta(n)$. A Las Vegas algorithm to embed any E -edge graph in $O(\sqrt{E})$ pages is also given.
- (2) The class of genus g graphs has pagenumber $O(\sqrt{g})$. This verifies a conjecture due to Heath and Istrail [HI]. The result is tight since the complete graph on n vertices has genus $\Theta(n^2)$ and pagenumber $\Theta(n)$. We give a Las Vegas algorithm to produce an $O(\sqrt{g})$ -page embedding when both the graph and its surface embedding are provided as input.

- (3) The $n \times n$ Mesh of Cliques has pagewidth $O(n^{3/2})$. The $n \times n$ Mesh of Cliques is the graph whose vertex set is $\{1, 2, \dots, n\} \times \{1, 2, \dots, n\}$ and whose edges connect each row $\{i\} \times \{1, 2, \dots, n\}$ into an n -vertex clique and each column $\{1, 2, \dots, n\} \times \{i\}$ into an n -vertex clique. The Mesh of Cliques was mentioned by Chung, Leighton, and Rosenberg [CLR] as a particularly nice example of a regular graph with unknown pagewidth.
- (4) Most n -vertex d -regular graphs have pagewidth $\Omega(\sqrt{dn}^{1/2-1/d})$. This bound is tight when $d > \log n$. The crux of the argument is a lemma that shows most d -regular graphs are "nearly" complete when suitably large clusters of vertices are identified. We suspect this lemma may have application to a variety of other graph embedding problems. For instance, we use it to derive an area lower bound for a multilayer VLSI grid model introduced by Aggarwal, Klawe, and Shor [AgKS].

Thesis Supervisor: Tom Leighton

Title: Associate Professor of Applied Mathematics



Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By <i>etc. on file</i>	
Distribution	
Availability Codes	
Dist	Avail and/or Special
A-1	

Acknowledgements

I would like to thank my advisor, Tom Leighton, for his energy, patience, and encouragement, all of which played a major role in the success of this thesis. He has been a tremendous source of inspiration for me, and I gained a great deal from his expertise. I would like to thank him for all his help in getting my career started, and for his concern over my academic and general well-being.

My appreciation goes to Lance Fortnow, Su Ming Wu, Jon Riecke, Toni Cormen, James Park, Joe Killian, Phil Klein, Charu, Sharad, and Raghu for valuable computer assistance. I would also like to thank Jerry Malitz, Sorin Istrail, Arny Rosenberg, Bill Aiello, Bruce Maggs, Cliff Stein, and Joel Wein for a variety of helpful discussions. This research was supported by NSF grants MIP83-52349 and CCR86-02062, Army Fellowship BAAL03-86-G-0015, Air Force contract OSR-86-0078, and DARPA contract N00014-87-K-825.

All my life, I have had the love, support, and encouragement of two wonderful people whom I dearly love, my parents Susan and Jerry. I never could have come this far without them. They have given me never-ending encouragement and inspiration to be the best that I can, and their enthusiasm for so many things has been one of the most profound positive influences in my life.

Lots of love goes to my younger brother Jed who has been a great brother and friend to me. I appreciate all his encouragement and understanding.

I would like to thank my good friend and colleague Ed Keppelmann for many great conversations, mathematical and otherwise, and lots of fun times.

During my stay at MIT, I was fortunate enough to befriend many wonderful people. They made the social scene at MIT lively and enjoyable, and I will miss them all very much. There are two people I would like to single out for their tremendous caring. These are Bill Aiello and Daniel Gatién. I also want to thank them for many entertaining conversations, and for sharing their knowledge of so many things.

My appreciation goes to Dick Holley, Dan Strook, James Cooper, and Irving Wiess for writing me letters of recommendation when I was applying for graduate school. In addition, I would like to thank Dick and Frieda Holley, and Dan and Lucy Strook for their freindship outside the academic realm.

I am grateful to the members of my thesis committee (Tom Leighton, Mike Sipser, Anders Bjorner, and David Schmoys) for all their time and effort.

As a graduate student, I had the opportunity to work with three excellent advisors: Albert Meyer, Shafi Goldwasser, and Tom Leighton. I am indebted to them all for their energy, patience and encouragement.

Lots of love and appreciation goes to my inlaws, Ann and Jerry, who have treated me with incredible kindness since the first day I met them.

Finally, I would like to thank my beautiful wife Jodi for her love, support, and understanding while I was writing this thesis. Her sweet and vivacious manner has added a terrific sparkle to my life, and there are so many things we share. I love her very much, and I am looking forward to our bright post-thesis future together.

For
the Malitz's,
the Goffman's,
and the Kingdom of Plants

Contents

1	Introduction	9
1.1	Definitions and Statement of the Problem	9
1.2	Circle Embeddings	10
1.3	Motivations	12
1.3.1	Multilayer Network Embedding	12
1.3.2	Design of Fault-Tolerant Processor Arrays	13
1.3.3	Sorting with Parallel Stacks	14
1.3.4	Turing Machine Graphs	14
1.4	Previous Work	15
1.5	Our Results	16
2	Embedding E-edge Graphs in $O(\sqrt{E})$ Pages	18
2.1	Introduction	18
2.2	Graphs with E Edges have Pagewidth $O(\sqrt{E})$	18
2.3	An Approximation Algorithm for an NP-Complete Problem	22
2.4	A Las Vegas Algorithm that Embeds an E -Edge Graph in $O(\sqrt{E})$ Pages	23
3	Genus g Graphs have Pagewidth $O(\sqrt{g})$	24
3.1	Introduction	24

3.2	Embedding Genus g Graphs in $O(g)$ Pages: The Heath-Istrail Algorithm	25
3.2.1	Introduction	25
3.2.2	The Decomposition	27
3.2.3	The Book-Embedding Algorithm	41
3.2.4	Some Additional Definitions	53
3.3	The Chain Graph Lemma	54
3.4	The One-Block Case	62
3.5	The General Case	66
3.6	A Las Vegas Algorithm	72
4	An Embedding for the Mesh of Cliques	74
4.1	Introduction	74
4.2	The Embedding	75
4.3	Comments	76
5	Two Lower Bounds for Regular Graphs	78
5.1	Introduction	78
5.2	Pagenumber Lower Bound for the Class of d -regular Graphs	79
5.3	An Area Lower Bound for Multilayer VLSI	80
6	Conclusion and Open Questions	85
6.1	Summary of the Results	85
6.2	Open Problems	86
	Index	89
	Bibliography	91

Chapter 1

Introduction

In this thesis, we prove several results about embedding graphs in **books**. The present chapter defines the **book embedding problem**, and indicates how it arises in VLSI design and parallel sorting. It also reviews what was known about book embeddings before our work, and outlines the results we have obtained.

1.1 Definitions and Statement of the Problem

Since we will be talking about embedding graphs in books, it is our first order of priority to define what we mean by a “graph”, a “book”, and embeddings of the former in the latter. A simple, undirected **graph** $G = (V, \mathcal{E})$ is a finite set V of vertices, and a set \mathcal{E} of edges which are two-element subsets of V . Thus a graph has no loops or multiple edges. A **book** is a two-part object consisting of a **spine**, which is a line, and some number of **pages**, each of which is a half-plane having the spine as boundary. A **book embedding** of a graph linearly orders the vertices along the spine of a book and assigns each edge to a page, so that edges assigned to the same page can be drawn on the page without crossings. Two book embeddings for the graph in Figure 1.1(a) are shown in Figures 1.1(b) and 1.1(c). Figure 1.1(b) is a 3-page embedding (thin edges go on one page, bold edges on another, and

patterned edges on the third), and Figure 1.1(c) is a 2-page embedding.

Clearly, a fixed graph G has many possible book embeddings. Define the **pagenumber** of an embedding to be the number of pages used in the book. Define the **pagenumber** of G to be the minimum pagenumber of any embedding of G . Define the **pagenumber** of a class of graphs to be the minimum number of pages in which every member of the class can be embedded, as a function of graph size. The **book embedding problem** is to find book embeddings with “small” pagenumber for all the members in a given graph family.

1.2 Circle Embeddings

There is an equivalent formulation of book embedding that is more convenient to use in certain situations, and this is called a **circle embedding**. A **circle embedding** of a graph G places the vertices at distinct locations on a circle (so that edges become chords), and assigns each chord to a layer so that chords in the same layer do not cross.

Not much effort is required to see that a book embedding with k pages is equivalent to a circle embedding with k layers. To obtain a circle embedding from a book embedding, simply take the spine of the book and “pull up” on the ends to form a circle. Two edges on the same page of the book embedding now become noncrossing chords of the circle, and thus can be placed on the same layer.

To obtain a book embedding from a circle embedding, cut the circle between two neighboring vertices (on the circle) and “pull down” on the ends to form the spine of a book. Two chords in the same layer of the circle embedding can now be placed on the same page of the book.

The equivalence of these two notions gives us the following simple facts: (1) G has a one-page book embedding iff it is outerplanar; (2) G has two-page book embedding iff it is a subgraph of planar graph that has a Hamiltonian cycle.

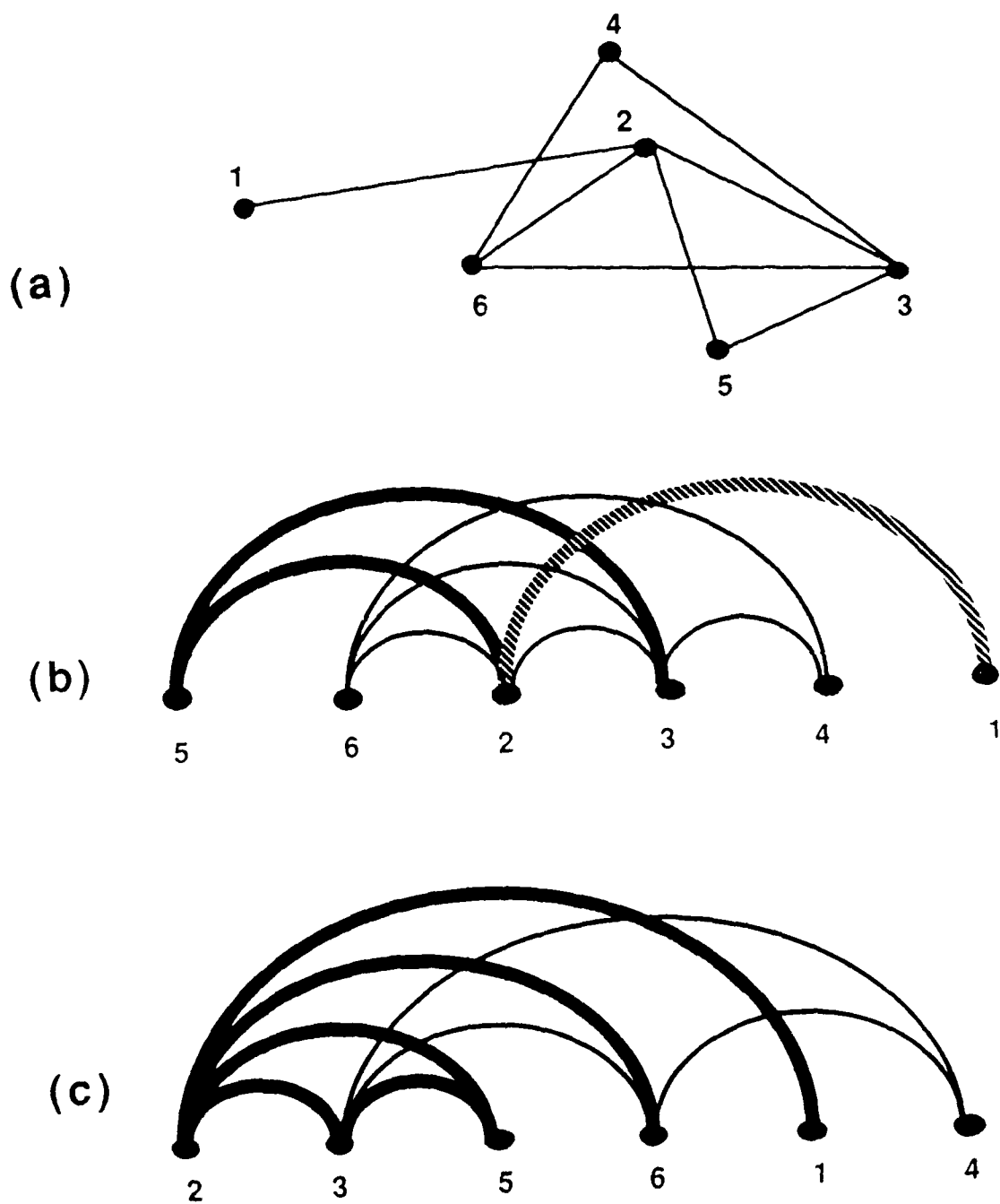


Figure 1.1: Two book embeddings for the graph in (a) are shown in (b) and (c).

1.3 Motivations

Book embeddings have application to several areas of theoretical computer science including VLSI design, automata theory, and complexity theory. We now briefly describe four of these applications. For more details see Heath [He1].

1.3.1 Multilayer Network Embedding

For the most part, VLSI layout theory has been a 2-dimensional theory of network embedding in 2-layer grids (see Leiserson [Ls].) Although two layers suffice, there are advantages to considering multilayer (more than two layer) grid embeddings, which takes the circuit layout problem into the realm of 3-dimensions. Leighton and Rosenberg ([LR1][LR2][Ro1][Ro2]) have studied 3-dimensional layout models and their results show that the volume and wire length of a good 3-dimensional layout are typically less than the area and wire length of the best 2-dimensional layout for the same circuit. On the practical side, multilayer printed circuit boards have been in existence for many years (So [So], Ting and Kuh [TK]), and recent technological advances are now making multilayer VLSI a reality.

The book embedding problem models a restricted version of 3-dimensional layout, where processors (that penetrate all the layers) are arranged in a conceptual line, and wires of the embedded network are prohibited from crossing this line, or changing layers. These restrictions arise naturally in multilayer channel routing and in the layout of multilayer printed circuit boards. Regarding the latter problem, for example, So [So] suggests a strategy called **Single-Row Routing**. Here the components of the network are arranged in a square array on a grid with evenly spaced channels separating the rows and columns. By adding dummy components if necessary, one can arrange for all the wires of the network to connect only components in the same row or column. The wires belonging to each row and column are then embedded independently in the space provided by the associated

channels. A 2-dimensional layout problem has thus been reduced to a collection of 1-dimensional layout problems, each each of which is solved independently. The restriction where wires are not allowed to pass between components in their row or column, corresponds precisely to the book embedding problem.

1.3.2 Design of Fault-Tolerant Processor Arrays

In the manufacturing process of computer chips, faulty processors are occasionally created, and these can render a chip unusable. One way of dealing with this problem is to engineer "customizable" chips, that can be wired later to form the desired network only among live processors. How might one design such a chip?

Rosenberg [Ro3] suggests a method called Diogenes. In this approach, the desired network is book-embedded in a 2-layer chip, and only live processors are implemented. The spine is a collection of identical processors arranged on the chip in a conceptual (if not actual) line, and there are enough of them to allow for expected failure. A page corresponds to a "bundle" of noncrossing wires that resides in the first layer and runs parallel to the spine. Different pages correspond to bundles located at different distances from the spine. Wires that run perpendicular to the spine connect the bundles to working processors. In the wiring process, we sweep across the spine from "left" to "right", incorporating the live processors into our desired network. This is being done for each bundle simultaneously. A simple switching mechanism built into the chip enables each wire-bundle to function as a hardware "stack" which is POPped when we interconnect a live processor that terminates a wire on the stack, and PUSHed when we interconnect a live processor that originates a wire on the stack. More precisely, when a functioning processor is encountered, a control signal is emitted by the processor perpendicularly through the bundle, indicating to all the wires, which stack operation is to be performed at that processor. If the stack is POPped, all the wires in the bundle shift one track closer to the spine and the inside wire interconnects the processor. If the stack is

PUSHed, all the wires shift one track further from the spine, thereby making room for a new wire interconnecting the processor. If a processor is faulty, no control signal is emitted, and the wires proceed on their present course until they arrive at the next processor. The beauty in this whole approach is that all wires in the same layer are treated identically each time a new processor is encountered. This makes for a very easy-to-implement wiring procedure.

Notice that book embeddings which use few pages correspond to reduced hardware (few layers). Since the Diogenes methodology takes as input an *arbitrary* interconnection graph, it is desirable to have good bounds on pagenumber for large classes of graphs.

1.3.3 Sorting with Parallel Stacks

Even and Itai [EI], Rosenstiehl and Tarjan [RT], and Tarjan [Ta], all studied the problem of realizing permutations with a collection of noncommunicating stacks. Let π be a permutation of the numbers $1, \dots, N$. Initially, each number in the order $1, \dots, N$ is PUSHed onto any one of k stacks, and then the stacks are POPped, yielding the sequence $\pi(1), \dots, \pi(N)$. Of course, this is only possible if $\pi(i)$ is on top of its stack after $\pi(1), \dots, \pi(i-1)$ are POPped. Suppose we want to know how many stacks are needed to realize π . Construct the graph G_π on vertices $\{a_1, \dots, a_N, b_1, \dots, b_N\}$ whose edges are $\{(a_i, b_{\pi(i)}) | 1 \leq i \leq N\}$. The minimum number of stacks required to realize π is precisely the pagenumber of G_π with respect to the vertex order $a_1, \dots, a_N, b_1, \dots, b_N$. This is because a page of G_π corresponds to numbers that are oppositely ordered by π . This means they can be PUSHed onto the same stack, since they will appear in the right order when the stack is POPped.

1.3.4 Turing Machine Graphs

Given a Turing machine M , one can construct a T -vertex graph that models T steps of M 's computation. Each vertex of the graph corresponds to a step of the

computation, and two vertices t_1 and t_2 are adjacent iff one of M 's tape heads visits the same square at times t_1 and t_2 , but at no time in between. It is not difficult to show that a k -tape Turing machine graph can be embedded in a $2k$ -page book. Some interesting results in complexity theory might be obtained by characterizing graphs that can be embedded in books with a given number of pages. For example, a proof that 3-page graphs have small bisection width would yield a number of interesting complexity-theoretic results. (See [GKS], [Ka], [PPST]).

1.4 Previous Work

The first important work on book embeddings is due to Bernhart and Kainen [BK]. They characterize one- and two-page embeddable graphs and demonstrate the equivalence of book embeddings and circle embeddings. They show that K_n , the complete graph on n vertices, has pagewidth $\lceil n/2 \rceil$. They establish the following relationship between the chromatic number of a graph and its pagewidth:

$$\chi(G) \leq 2 \cdot \text{page}(G) + 2.$$

They also construct graphs with pagewidth 3 and arbitrarily large genus (see Subsection 3.2.1 for the definition of graph genus.) In the other direction, they conjectured that graphs of fixed genus could require an unbounded number of pages.

For genus 0 graphs (i.e. planar graphs), this conjecture was disproved by Buss and Shor [BS] who showed how to embed any planar graph in 9 pages. Using other techniques, Heath [He1] improved that to 7 pages and developed book embedding algorithms for special classes of planar graphs [He2]. Later, Istrail [Is] showed that planar graphs require no more than 6 pages. Finally, extending the techniques of Heath [He2], Yannakakis [Ya] proved that 4 pages are necessary and sufficient for the class of planar graphs. An important aspect of this 4-page embedding, is that for a biconnected planar graph, it preserves the cyclic order of the boundary vertices (i.e. the vertices bounding the exterior face.)

The general conjecture for arbitrary fixed genus was disproved by Heath and Istrail [HI] who provide an algorithm to embed any genus g graph in $O(g)$ pages. (Their algorithm is linear-time if the surface embedding is given as part of the input.) They also demonstrate that $\Omega(\sqrt{g})$ pages are necessary for the class of genus g graphs. This led them to conjecture that $O(\sqrt{g})$ pages would suffice for the class.

Another important work on book embeddings pursues some different directions. Chung, Leighton, and Rosenberg [CLR] give optimal book embeddings for a variety of popular networks, present a polynomial-time algorithm to embed trivalent graphs in $O(\sqrt{n})$ pages, and relate pagewidth to the size of a graph's minimum bifurcator (see [BL]). In addition, they demonstrate that the class of d -regular graphs on n nodes has pagewidth $O(d\sqrt{n})$ and $\Omega(n^{1/2-1/d} \log^2 n)$.

From a computational point of view, determining the pagewidth of an arbitrary graph appears difficult. Garey, Johnson, Miller and Papadimitriou [GJMP] have shown that even if an ordering of the vertices is specified beforehand, determining the optimal number of pages for general graphs is NP-complete. Equally interesting, Wigderson [Wi] has shown that to decide whether or not a planar graph has pagewidth 2 is NP-complete.

1.5 Our Results

Chapter 2 extends the ideas in [CLR] to show that the class of E -edge graphs has pagewidth $O(\sqrt{E})$. This result is tight since the complete graph on n vertices has $\Theta(n^2)$ edges and pagewidth $\Theta(n)$. A Las Vegas algorithm to embed an E -edge graph in $O(\sqrt{E})$ pages is also given.

Chapter 3 goes even further, extending the result above and the result of [HI]. Here we show that the class of genus g graphs has pagewidth $O(\sqrt{g})$, thus verifying the Heath-Istrail Conjecture. This bound is tight since the complete graph on

n vertices has genus $\Theta(n^2)$ and pagewidth $\Theta(n)$. We give a Las Vegas algorithm to embed a genus g graph in $O(\sqrt{g})$ pages when both the graph and its surface embedding are provided as input. Since the Heath-Istrail algorithm [HI] is a fundamental building block for our embedding strategy, we review the necessary results from their paper in Section 3.2. To make the thesis more self-contained, all proofs of their results have been included.

The $n \times n$ **Mesh of Cliques**, $M(n)$, is the graph whose vertex set is $\{1, 2, \dots, n\} \times \{1, 2, \dots, n\}$ and whose edges connect each row $\{i\} \times \{1, 2, \dots, n\}$ into an n -vertex clique and each column $\{1, 2, \dots, n\} \times \{i\}$ into an n -vertex clique. The Mesh of Cliques was mentioned by Chung, Leighton, and Rosenberg [CLR] as a particularly nice example of a regular graph with unknown pagewidth. They show (nonconstructively) that $M(n)$ has pagewidth $O(n^2)$, and that any book embedding which places $M(n)$'s vertices along the spine row-by-row must use $n^{4/3}$ pages. In Chapter 4, we give a constructive $O(n^{3/2})$ -page embedding for $M(n)$ that maintains a row-by-row ordering of the vertices along the spine.

In Chapter 5, we prove that most n -vertex d -regular graphs have pagewidth $\Omega(\sqrt{dn}^{1/2-1/d})$. This strengthens the result of [CLR], and is tight for $d > \log n$. The crux of the argument is a lemma that shows most d -regular graphs are "nearly" complete when suitably large clusters of vertices are identified. We suspect this lemma may have application to a variety of other graph embedding problems. In Section 5.3, for instance, it is used to derive an area lower bound for a multilayer VLSI grid model introduced by Aggarwal, Klawe, and Shor [AgKS].

Chapter 6 summarizes our results and closes with a list of open problems.

Chapter 2

Embedding E -edge Graphs in $O(\sqrt{E})$ Pages

2.1 Introduction

In Section 2.2, we extend the ideas of [CLR] to show that the class of E -edge graphs has pagewidth $O(\sqrt{E})$. In Section 2.3, we give an approximation algorithm for the NP-complete problem of optimally book embedding a graph with respect to a specified ordering of the vertices. Finally in Section 2.4, we supply a Las Vegas algorithm to embed any graph with E edges in $O(\sqrt{E})$ pages.

2.2 Graphs with E Edges have Pagewidth $O(\sqrt{E})$

The following is a precise statement of the theorem we wish to establish.

Theorem 2.2.1 Let G be a graph with E edges on n nodes, where $E \geq n$. Given a random linear ordering π of the vertices, the probability that G can be embedded in $11\sqrt{E}$ pages with respect to the ordering π , is at least $1 - (1/2)^{\sqrt{n}}$.

In order to prove this, we need a few definitions and preliminary lemmas.

A **2-colored bipartite graph** (2BG) is a bipartite graph with an associated 2-coloring of the vertices, say with the colors LEFT and RIGHT, so that no edge connects two vertices of the same color. Given a bipartite graph G with k connected components, there are precisely 2^k ways of 2-coloring the vertices with the colors LEFT and RIGHT.

A 2BG is **canonically ordered** if its vertices are linearly ordered so that all LEFT vertices precede all RIGHT vertices.

Given a (horizontal) linear ordering of the vertices in an arbitrary 2BG, and an edge e in the 2BG, denote the left and right endpoints of e by $l(e)$ and $r(e)$ respectively.

A 2-colored matching is **completely crossing** with respect to the canonical ordering π , if its edges can be labeled e_1, e_2, \dots, e_k so that

$$l(e_1) <_{\pi} l(e_2) <_{\pi} \dots <_{\pi} l(e_k) \text{ and } r(e_1) <_{\pi} r(e_2) <_{\pi} \dots <_{\pi} r(e_k).$$

Intuitively, a 2-colored matching is completely crossing with respect to π , if no edge can be placed on the same page as another edge.

The following lemma generalizes a result of Tarjan [Ta] (Theorem 1.2 in [CLR]) and the proof is essentially identical.

Lemma 2.2.1. Let π canonically order the 2BG G . If at most k edges are completely crossing with respect to π , then G can be embedded in at most k pages with respect to π .

Proof. Partially order the edges of G as follows. Say

$$e_1 \preceq e_2 \text{ iff } [l(e_2) \leq_{\pi} l(e_1) \text{ and } r(e_1) \leq_{\pi} r(e_2)].$$

Intuitively $e_1 \preceq e_2$ iff e_1 is "nested" in e_2 with respect to π . A chain in this partial order corresponds to a set of edges that can be placed on the same page, and an antichain corresponds to a set of completely crossing edges. Dilworth's Theorem

(see [Di]) says that the minimal number of chains into which a partial order can be decomposed is equal to the size of the largest antichain. ■

Lemma 2.2.2. [CLR] For n even, K_n has pagenumber $n/2$.

Proof. The lower bound on pagenumber is argued as follows. Arbitrarily lay the vertices of K_n out on a line; call the vertices $1, \dots, n$ in left-to-right order. Consider the set of edges $\{(1, 1 + n/2), (2, 2 + n/2), \dots, (n/2, n)\}$, and observe that no pair of edges from this collection can be placed on the same page without intersecting. Hence this embedding requires $n/2$ pages.

To see the upper bound, consider the following way to lay out K_n . Place the vertices $1, \dots, n$ evenly spaced on a circle and look at the path P_n shown below:

$$P_n = 1 - n - 2 - (n-1) - 3 - (n-2) - \dots - n/2 - (n - n/2 + 1).$$

If we take the union of P_n and $(n/2) - 1$ consecutive clockwise rotates of P_n , we obtain K_n . In fact, one can verify that each edge of K_n appears in exactly one of the $n/2$ rotates of P_n . Since each rotate of P_n is outerplanar, we can place it on its own page, thereby obtaining an $n/2$ -page embedding for K_n . ■

We are now ready to establish Theorem 2.2.1. The proof combines the development above with ideas from Theorem 4.7 of [CLR].

Proof of Theorem 2.2.1. Pick a random linear ordering π of G 's vertices, and partition the edges of G into $\log n$ "levels" (all logs are base 2). To obtain the j -level edges ($1 \leq j \leq \log n$), first divide π into 2^j segments each with the same number of vertices. Label the segments from left to right L.R.L.R.L, etc. Any edge that connects vertices in an adjacent L.R pair of segments is a j -level edge. Notice that every edge of G is in a unique level.

Let A_k^j be the event that there exists a matching M in G with k edges, and a 2-coloring χ of M , such that the resulting 2BG is canonically ordered, j -level, and

completely crossing with respect to π . We have

$$Pr[A_k^j] < \underbrace{2^{j-1}}_1 \underbrace{\binom{E}{k}}_2 \underbrace{2^k \left(\frac{1}{2^j}\right)^{2k}}_3 \underbrace{\frac{1}{k!}}_4$$

where

- (1) is the number of adjacent L,R segment pairs at the j th level;
- (2) upper bounds the number of matchings M in G with k edges (including a designated 2-coloring);
- (3) upper bounds the probability that π canonically orders the vertices of M by putting all LEFT nodes in a fixed segment labeled L and all RIGHT nodes in the adjacent segment labeled R;
- (4) upper bounds the probability that M is completely crossing with respect to π , given that π canonically orders M .

Simplifying,

$$Pr[A_k^j] < \frac{E^k}{k!} \left(\frac{1}{\sqrt{2^j}}\right)^{2k} \frac{1}{k!} < \frac{e^{2k} E^k}{(k \sqrt{2^j})^{2k}}.$$

If we let $k_j = e \sqrt{2E} / \sqrt{2^j}$, then

$$Pr[A_{k_j}^j] < (1/2)^{e\sqrt{2E}/\sqrt{2^j}} < (1/2)^{e\sqrt{n}/\sqrt{2^j}}$$

where the last inequality follows from the assumption $E \geq n$. The next step is to show that simultaneously we expect no more than $e\sqrt{2E}/\sqrt{2^1}$ completely crossing 1-level edges, no more than $e\sqrt{2E}/\sqrt{2^2}$ completely crossing 2-level edges, no more than $e\sqrt{2E}/\sqrt{2^3}$ completely crossing 3-level edges, and so on in a nice geometrically decreasing fashion. We have

$$\begin{aligned} & Pr[A_{k_1}^1 \vee A_{k_2}^2 \vee \dots \vee A_{k_{\log(n/\sqrt{E})}}^{\log(n/\sqrt{E})}] \\ & \leq Pr[A_{k_1}^1 \vee A_{k_2}^2 \vee \dots \vee A_{k_{\log \sqrt{n}}}^{\log \sqrt{n}}] \end{aligned}$$

$$\begin{aligned}
&< (1/2)^{e\sqrt{n}/\sqrt{2}^1} + (1/2)^{e\sqrt{n}/\sqrt{2}^2} + \dots + (1/2)^{e\sqrt{n}} \\
&< (1/2)^{\sqrt[4]{n}}.
\end{aligned}$$

Hence

$$Pr[\neg A_{k_1}^1 \wedge \neg A_{k_2}^2 \wedge \dots \wedge \neg A_{k_{\log(n/\sqrt{E})}}^{\log(n/\sqrt{E})}] > 1 - (1/2)^{\sqrt[4]{n}}.$$

Therefore, with probability at least $1 - (1/2)^{\sqrt[4]{n}}$, a random linear ordering of the vertices allows the j -level edges to be embedded in $e\sqrt{2E}/\sqrt{2}^j$ pages (by Lemma 2.2.1) for all j in the range 1 through $\log(n/\sqrt{E})$. (We did not go all the way to level $\log n$ because as j approaches $\log n$, the quantity $Pr[A_{k_j}^j]$ approaches a constant, and this would not allow us to obtain a high probability result.)

Observe now that segments at level $\log(n/\sqrt{E})$ contain only \sqrt{E} vertices. Hence the remaining edges, those in levels $1 + \log(n/\sqrt{E})$ and beyond, can be embedded in \sqrt{E} pages by Lemma 2.2.2. Thus, for "good" linear orderings, the total number of pages used to embed all the edges of G is at most

$$e\sqrt{2E}/\sqrt{2} + e\sqrt{2E}/\sqrt{2}^2 + \dots + e\sqrt{2E}/\sqrt{2}^{\log(n/\sqrt{E})} + \sqrt{E} \leq 11\sqrt{E}. \blacksquare$$

2.3 An Approximation Algorithm for an NP-Complete Problem

In Section 1.4, we mentioned the following result of Gary et. al. [GJMP]: that it is NP-complete to optimally book embed a general graph with respect to a specified ordering of its vertices. In this section, we describe a simple log-factor approximation algorithm due to Tom Leighton [Lt3]. Given an input graph G on n nodes and a linear ordering L of its vertices, the algorithm embeds the edges of G in $\log n$ times optimal pages with respect to L , in deterministic polynomial time. The idea is to first decompose G into $\log n$ bipartite graphs, and then embed each bipartite graph in the optimal number of pages using an effective version of Lemma 2.2.1. (It

is well known that Dilworth's original proof of Dilworth's Theorem (see [Di]) can easily be adapted to a polynomial-time algorithm which takes a partial order and decomposes it into the minimum number of chains.)

The approximation algorithm works as follows. Partition the edges of G into levels with respect to L , as done in the proof of Theorem 2.2.1. Using the deterministic polynomial-time version of Lemma 2.2.1, embed all j -level edges in the optimal number of pages, say OPT_j . By employing a different set of pages for each level, the total number of pages utilized by the algorithm is $OPT_1 + OPT_2 + \dots + OPT_{|L|}$. If OPT is the smallest number of pages in which all of G can be embedded, then clearly $\max_j OPT_j \leq OPT$, and hence, the algorithm uses no more than $OPT \cdot \log n$ pages.

2.4 A Las Vegas Algorithm that Embeds an E -Edge Graph in $O(\sqrt{E})$ Pages

The Las Vegas algorithm simply applies the above routine to a random linear ordering of the vertices of the input graph. If more than $O(\sqrt{E})$ pages total are employed, the algorithm picks another random ordering and applies the routine again. It continues picking random orderings until the routine finally yields a book embedding with $O(\sqrt{E})$ pages. By the proof of Theorem 2.2.1, this algorithm runs in expected polynomial time.

Chapter 3

Genus g Graphs have Pagenumber $O(\sqrt{g})$

3.1 Introduction

The main result of this chapter is an extension of the results in [HI] and Chapter 2. We show that $O(\sqrt{g})$ pages do indeed suffice for the class of genus g graphs, thus verifying the Heath-Istrail Conjecture. As in Chapter 2, we also give a Las Vegas algorithm to embed a genus g graph in $O(\sqrt{g})$ pages.

There are 5 sections that follow. Section 3.2 reviews the crucial results we need from the paper by Heath and Istrail [HI]. To make the thesis more self-contained, proofs of all their results are included in the section. Section 3.3 introduces an important class of graph-like objects called "chain graphs" and establishes an upper bound on their pagenumbers. Chain graphs play a major role in our embedding strategy for genus g graphs. Section 3.4 derives the $O(\sqrt{g})$ upper bound on pagenumber for a simple and instructive class of genus g graphs. Section 3.5 derives the $O(\sqrt{g})$ upper bound for general genus g graphs. Finally in Section 3.6, we supply a Las Vegas algorithm to embed any genus g graph in $O(\sqrt{g})$ pages when both the graph and its initial surface embedding are provided as input.

3.2 Embedding Genus g Graphs in $O(g)$ Pages: The Heath-Istrail Algorithm

3.2.1 Introduction

The algorithm of Heath and Istrail [HI], which embeds a genus g graph in $O(g)$ pages, is a fundamental building block in our $O(\sqrt{g})$ -page embedding strategy. For the sake of completeness, we review their results in the present section, and include proofs of all the lemmas. We stress that there is no original research on our part contained in this section.

To begin, we need a few definitions. We follow the development of White [Wh] for graphs embedded in surfaces. An **orientable** surface of **genus** g is a sphere with g **handles**—a handle is a cylindrical tube whose ends are attached to the boundaries of two disks cut in the sphere. Figure 3.1(a) shows a sphere with two handles. A **nonorientable** surface of genus g is a sphere with g **cross-caps**, which are defined as follows. Continuously deform a cylindrical tube so that each point on one of its circular boundaries is mapped to the point diametrically opposite. This yields a surface that intersects itself in 3-space, but not in 4-space. We get a cross-cap by attaching this new object to the boundaries of two disks cut in the sphere. Figure 3.1(b) depicts a sphere with two cross-caps. A connected graph $G = (V, E)$ is **embedded** in a surface if it is drawn on the surface without crossing edges. The **genus** of G , $\gamma(G)$, is the minimum genus of an orientable surface into which G is embeddable. The **nonorientable genus** of G , $\gamma'(G)$, is analogous for nonorientable surfaces. Connected components of the complement of an embedding of G are the **faces** of the embedding. The embedding is **2-cell** if every face is homeomorphic to an open disk. It so happens, that any embedding of G in an orientable surface of genus $\gamma(G)$ is a 2-cell embedding. (See [Wh], Theorem 6.11.) The analogous statement for nonorientable surfaces is not necessarily true.

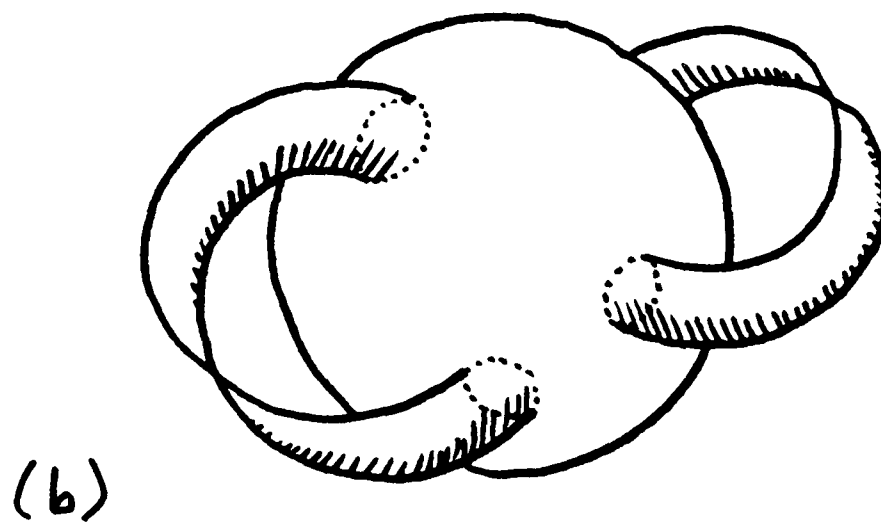
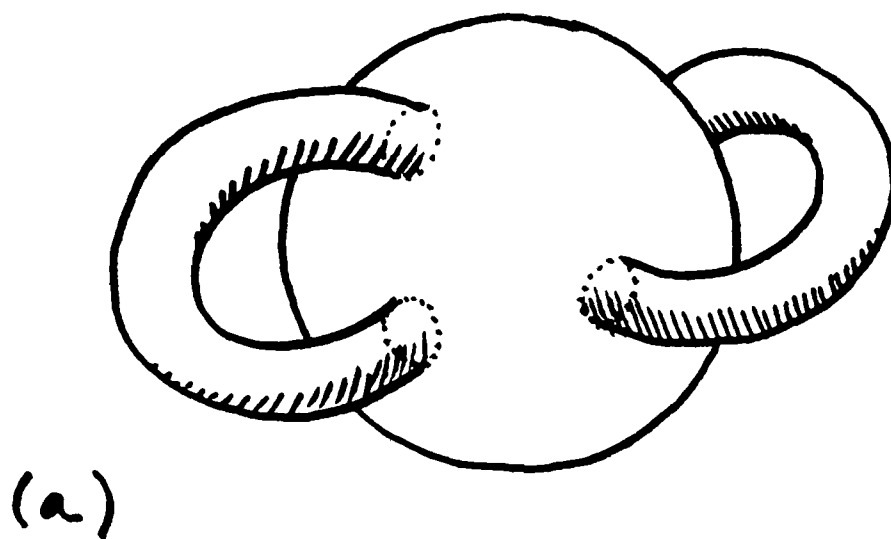


Figure 3.1: A sphere with two handles is shown in (a). A sphere with two cross-caps is shown in (b).

The Heath-Istrail algorithm starts initially with a connected n -node graph $G = (V, \mathcal{E})$ of genus g that is 2-cell embedded in a surface S of (orientable or nonorientable) genus g . (Such an embedding can be computed in time $O(n^{O(g)})$ by an algorithm of Filotti, Miller, and Rief [FMR].) G is then embedded in $O(g)$ pages by applying a linear-time algorithm, an important aspect of which is an interesting decomposition for a graph embedded on a surface.

Since our result that genus g graphs have pagewidth $O(\sqrt{g})$ is established only for *orientable* genus g , we will discuss the algorithm of Heath and Istrail that deals only with orientable surfaces. We leave it as an open question whether or not a graph of nonorientable genus g has an $O(\sqrt{g})$ -page embedding. From now on, when we speak of *genus* we will always mean orientable genus.

There are 3 subsections that follow. Subsection 3.2.2 develops the decomposition mentioned above, for graphs embedded on surfaces. The book embedding algorithm is given in Subsection 3.2.3. Finally Subsection 3.2.4 is reserved for two definitions that will be important later on.

3.2.2 The Decomposition

In order to define this decomposition, we need a combinatorial representation of graphs embedded in surfaces due to Edmunds [Ed]. For each $v \in V$, the **neighborhood** of v is $N(v) = \{u | (u, v) \in \mathcal{E}\}$. A **rotation** of G is a set of $|V|$ cyclic permutations, one for each vertex:

$$R = \{\sigma_v | v \in V \text{ and } \sigma_v \text{ is a cyclic permutation of } N(v)\}.$$

If $H = (V_H, \mathcal{E}_H)$ is a subgraph of G , define $N_H(v) = \{u | (u, v) \in \mathcal{E}_H\}$. If σ_v is a cyclic permutation of $N(v)$, then define $\sigma_{v,H}$ to be the cyclic permutation of $N_H(v)$ that is consistent with the cyclic order of σ_v . A rotation R of G induces the **subrotation** $R_H = \{\sigma_{v,H} | v \in V_H\}$ of H .

Rotations can be used to represent surface embeddings. If G has a 2-cell embedding in an orientable surface, there is a rotation R representing this embedding:

σ_v is given by examining the vertices adjacent to v , in clockwise order about v (on the surface).

A **planar-nonplanar decomposition** of $G = (V, \mathcal{E})$ is a triple (R, G_P, \mathcal{E}_N) that satisfies three properties. Here R is a rotation representing a 2-cell embedding of G into an orientable surface, $G_P = (V, \mathcal{E}_P)$ is a planar subgraph of G , and $\mathcal{E}_N = \mathcal{E} - \mathcal{E}_P$. The properties satisfied are

- (1) The subrotation R_{G_P} induces a planar embedding of G_P .
- (2) Every edge of \mathcal{E}_N has both endpoints on the exterior face of G_P .
- (3) \mathcal{E}_P is maximal, i.e. no edge of \mathcal{E}_N can be added to G_P without violating either property (1) or (2).

The edges in \mathcal{E}_N are referred to as the **nonplanar** edges of the decomposition. For example, given the graph in Figure 3.2(a), a choice for G_P is shown in Figure 3.2(b). The nonplanar edges are (v_1, v_4) and (v_3, v_5) .

Take F_0 to be the exterior face of the planar embedding G_P . We can imagine traversing the boundary of F_0 in, say, clockwise order. This defines a directed cycle (which is typically not simple as G_P may have articulation points and cut-edges.) A directed subpath of this directed cycle is a **trace**. If the trace T consists of v_1, v_2, \dots, v_t , then denote the trace by

$$T = v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_t.$$

The **endpoints** of T are v_1 and v_t . The **inverse trace** to T is

$$T^{-1} = v_t \rightarrow v_{t-1} \rightarrow \dots \rightarrow v_1,$$

i.e. the trace obtained by traversing T in the reverse direction. In Figure 3.3 for example,

$$w_4 \rightarrow w_5 \rightarrow w_6 \rightarrow w_7 \rightarrow w_8 \rightarrow w_5$$

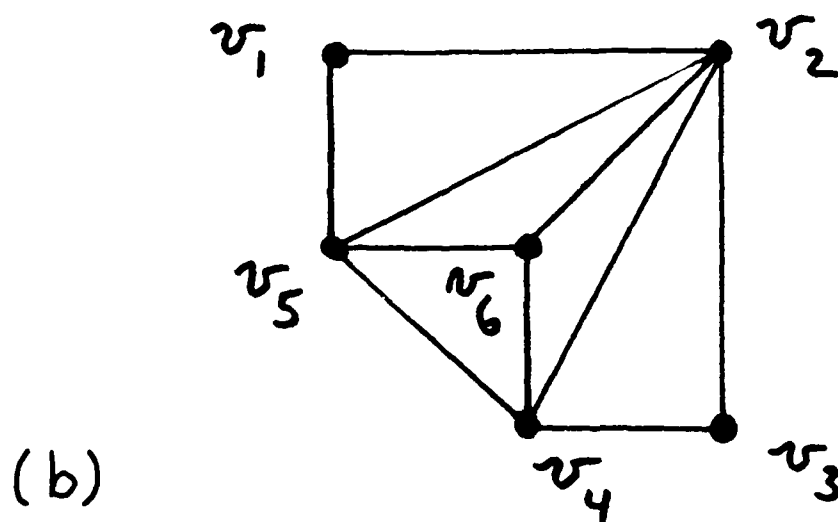
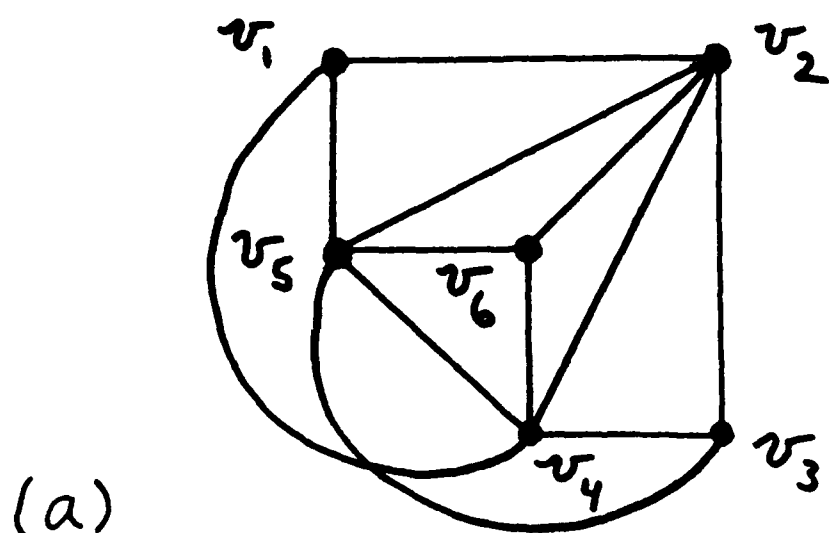


Figure 3.2: Given the graph in (a), a choice for G_F is shown in (b). The nonplanar edges are (v_1, v_4) and (v_3, v_5) .

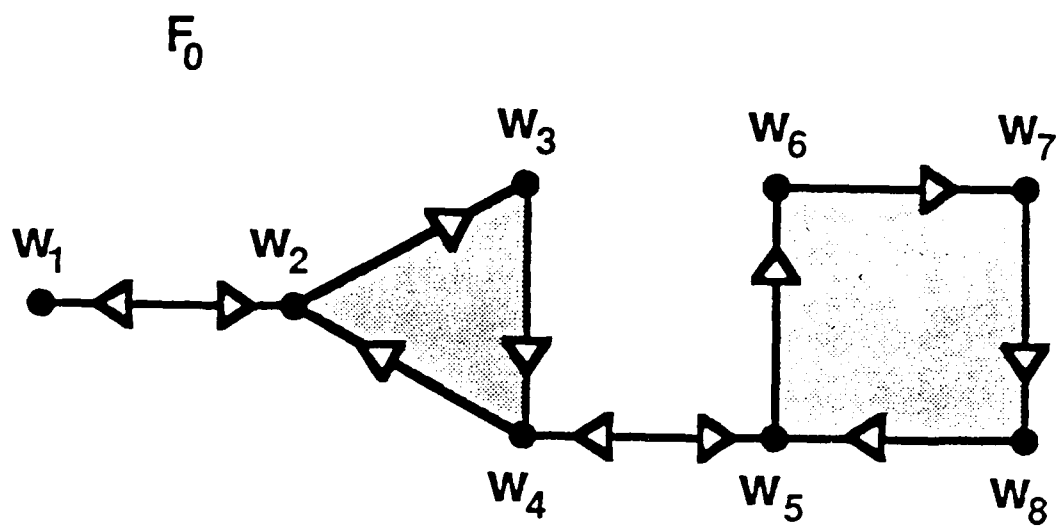


Figure 3.3: Traversal of the boundary of the planar part. (Taken from [HI]).

is a trace with inverse trace

$$w_5 \rightarrow w_8 \rightarrow w_7 \rightarrow w_6 \rightarrow w_5 \rightarrow w_4.$$

The path

$$w_2 \rightarrow w_3 \rightarrow w_4 \rightarrow w_2$$

is not a trace because it does not follow a clockwise or counterclockwise traversal of the boundary of G_P .

Given a planar-nonplanar decomposition (R, G_P, \mathcal{E}_N) of G , we can partition \mathcal{E}_N into equivalence classes. Suppose that $(u_1, v_1), (u_2, v_2) \in \mathcal{E}_N$ are part of the boundary of the same face F of the surface embedding of G , and that F is not a face of G_P . Then (u_1, v_1) and (u_2, v_2) are **homotopic** [written $(u_1, v_1) \sim_h (u_2, v_2)$] if

- (1) (u_1, v_1) and (u_2, v_2) are the only edges of \mathcal{E}_N on the boundary of F ;
- (2) there are traces $T_{u_i} = u_1 \rightarrow \dots \rightarrow u_2$ and $T_{v_i} = v_1 \rightarrow \dots \rightarrow v_2$ such that both T_{u_i} and T_{v_i} lie on the boundary of F .

If (u_1, v_1) and (u_2, v_2) are homotopic, then the entire boundary of F consists of $(u_1, v_1), (u_2, v_2), T_{u_i}$, and T_{v_i} . The relation \equiv_h is defined to be the reflexive, symmetric, transitive closure of \sim_h . Clearly \equiv_h is an equivalence relation on \mathcal{E}_N . Each equivalence class is called a **homotopy class**. (Notice that a homotopy class can consist of a single edge.) Consider Figure 3.4. Here the oval represents the boundary of G_P , and the bold lines represent nonplanar edges. The three homotopy classes are

$$\{(u_1, v_1), (u_2, v_2), (u_3, v_3)\},$$

$$\{(u_4, v_4), (u_5, v_5), (u_6, v_6)\},$$

and

$$\{(u_7, v_7), (u_8, v_8), (u_9, v_9)\}.$$

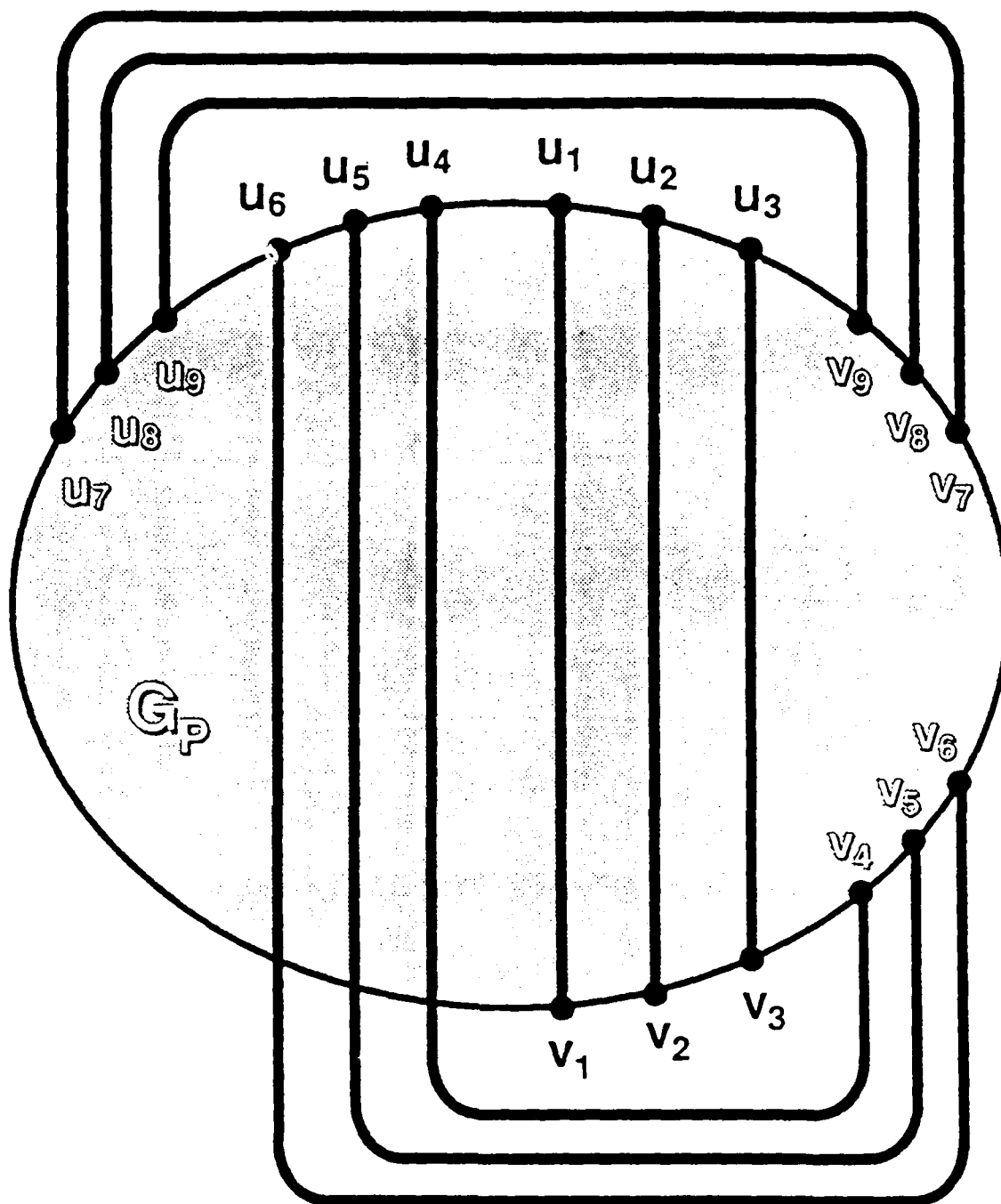


Figure 3.4: Homotopy classes. (Taken from [HI]).

In all the lemmas that follow, let G be a graph 2-cell embedded in an orientable surface S .

Lemma 3.2.1. [HI] If C is a homotopy class of G , then the elements of C can be ordered

$$(u_1, v_1), \dots, (u_k, v_k)$$

and two traces T_1 and T_2 defined such that

- (1) for $1 \leq i \leq k-1$, (u_i, v_i) is homotopic to (u_{i+1}, v_{i+1}) with corresponding traces T_{u_i} and T_{v_i} ;
- (2) T_1 is the concatenation of

$$T_{u_1}, T_{u_2}, \dots, T_{u_{k-1}}$$

and T_2 is the concatenation of

$$T_{v_1}, T_{v_2}, \dots, T_{v_{k-1}}.$$

Note that u_i, u_{i+1} need not be distinct, and v_i, v_{i+1} need not be distinct. However, $u_i = u_{i+1}$ and $v_i = v_{i+1}$ is not allowed as G may not have multiple edges.

Proof. An enumeration that satisfies (1) is clearly possible since every edge of a 2-cell embedding is incident to only two faces. We must show that such an enumeration yields a T_1 and T_2 that are *traces*.

To show this, we want to make a simplifying assumption that needs a little argument. We want to show that without loss of generality, we can assume any two homotopic edges of C determine a triangular face. Fix a nontriangular face F of G bounded by two homotopic edges $(u_j, v_j), (u_{j+1}, v_{j+1}) \in C$ and the corresponding traces T_{u_j}, T_{v_j} . Create a new graph G' by adding an edge (μ, ν) (distinct from (u_j, v_j) and (u_{j+1}, v_{j+1})) inside F that spans T_{u_j} and T_{v_j} . Then (u_j, v_j) is homotopic

to (μ, ν) with corresponding traces T'_{u_j}, T'_{v_j} , and (μ, ν) is homotopic to (u_{j+1}, v_{j+1}) with corresponding traces T_μ, T_ν . Hence we can *extend* the decomposition of G to a decomposition of G' for which $C' = C \cup \{(\mu, \nu)\}$ is a homotopy class, and notice that if we concatenate the associated traces of C' we get exactly T_1 and T_2 . No other homotopy class of G is affected by the addition of (μ, ν) . Thus without loss of generality, we may assume that any two homotopic edges of C determine a triangular face.

Let $R = \{\sigma_w | w \in V\}$ be the (clockwise) rotation of G representing its 2-cell embedding. Suppose (u_{i-1}, v_{i-1}) is homotopic to (u_i, v_i) which is homotopic to (u_{i+1}, v_{i+1}) , and let us assume for contradiction that $u_{i-1} \rightarrow u_i \rightarrow u_{i+1}$ is *not* a trace and $u_{i-1} \neq u_i \neq u_{i+1}$. Since $v_{i-1} = v_i = v_{i+1}$, we will denote them all by v . We can immediately rule out the possibility that $u_i \neq u_{i-1}$ and $u_{i+1} = u_{i-1}$ since in this case G would have multiple edges. The only other possibility to consider is the case where u_i is an articulation point of G_P , $u_i = u_{i-1}$, and $u_{i+1} = u_i, u_{i-1}$. (See Figure 3.5.) Starting with the edge $(u_i, u_{i-1}) \in G_P$, imagine sweeping clockwise about u_i in the plane. Since $u_{i-1} \rightarrow u_i \rightarrow u_{i+1}$ is not a trace, we encounter an edge $(u_i, z) \in G_P$ before we encounter the edge $(u_i, u_{i+1}) \in G_P$. But the assumption that $(u_{i-1}, v_{i-1}) \rightarrow_h (u_i, v_i) \rightarrow_h (u_{i+1}, v_{i+1})$ implies that $\sigma_{u_i}(u_{i-1}) = v$ and $\sigma_{u_i}(v) = u_{i+1}$. Hence, to be consistent with the rotation R , u_{i+1} should be the next edge of G_P immediately clockwise from (u_i, u_{i-1}) about u_i . Figure 6 is therefore an impossibility. ■

Lemma 3.2.2. [HI] Traces from different homotopy classes do not intersect except possibly at endpoints and articulation points of G_P .

Proof. Let T_1 and T_2 be traces from different homotopy classes. If T_2 shares a vertex v interior to T_1 , and v is not an articulation point of G_P , then T_2 shares an interior edge of T_1 . Such an edge would be incident to three faces—one in G_P , and two outside G_P . But in a 2-cell embedding, every edge is incident to precisely two faces. Thus T_2 can only share vertices of the specified kind with T_1 . ■

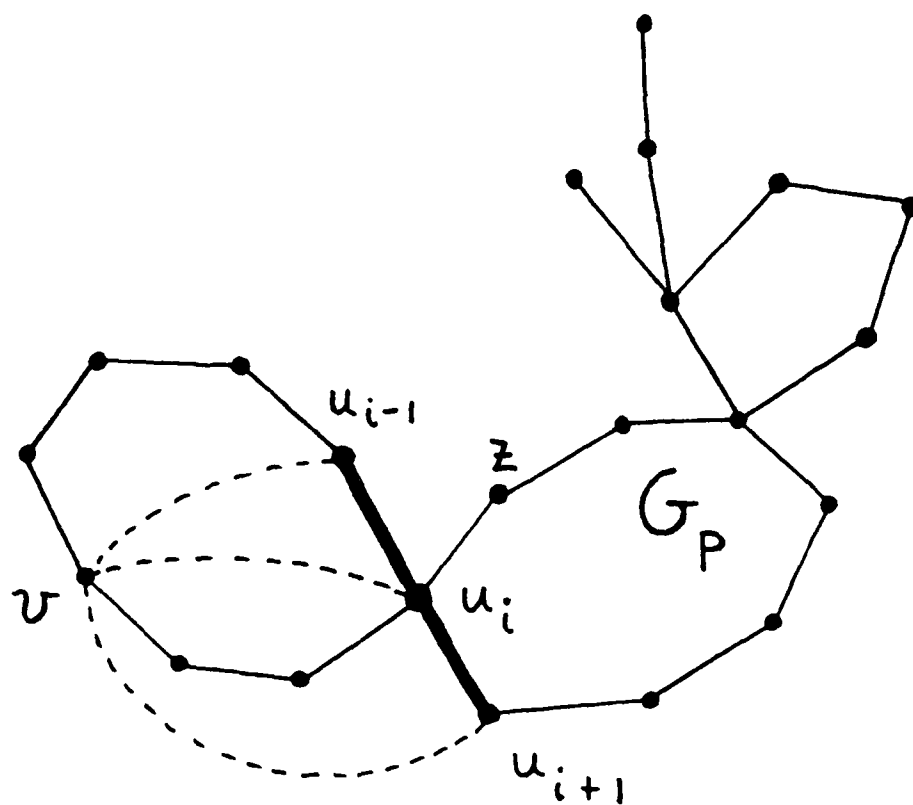


Figure 3.5: Construction for Lemma 3.2.1.

Figure 3.6 shows the possible trace intersections for different homotopy classes.

As the boundary of the planar graph is traversed in a consistent direction, each of the two traces of a homotopy class is encountered exactly once. When the traces are traversed in opposite orders, the homotopy class is **orientable**. When the two traces are traversed in the same order, the homotopy class is **nonorientable**. In Figure 3.4, the homotopy classes

$$\{(u_1, v_1), (u_2, v_2), (u_3, v_3)\}$$

and

$$\{(u_7, v_7), (u_8, v_8), (u_9, v_9)\}$$

are orientable, whereas the homotopy class

$$\{(u_4, v_4), (u_5, v_5), (u_6, v_6)\}$$

is nonorientable.

Lemma 3.2.3. [HI] Every homotopy class of any planar-nonplanar decomposition of G is orientable.

Proof. Fix a planar-nonplanar decomposition of G , and let C be a homotopy class of the decomposition. As in the proof of Lemma 3.2.1, we may assume without loss of generality, that every pair of homotopic edges in C determines a triangular face.

Suppose, for contradiction, that C is a nonorientable homotopy class. Let its traces be T_1 and T_2 (both traces directed clockwise, say, about the boundary of G_P). Neither trace consists of just a single vertex, otherwise C is orientable. Hence, there is a subtrace $u_i \rightarrow u_{i+1}$, $u_i \neq u_{i+1}$ of T_1 and a subtrace $v_j \rightarrow v_{j+1}$, $v_j \neq v_{j+1}$ of T_2 such that $(u_i, v_j), (u_{i+1}, v_{j+1}) \in C$ and such that either: (1) $(u_i, v_{j+1}) \in C$ and $(u_i, v_j) \rightarrow_h (u_i, v_{j+1}) \rightarrow_h (u_{i+1}, v_{j+1})$, or (2) $(u_{i+1}, v_j) \in C$ and $(u_i, v_j) \rightarrow_h (u_{i+1}, v_j) \rightarrow_h (u_{i+1}, v_{j+1})$. Without loss of generality, say (1) holds. (See Figure 3.7.) Then the cycle

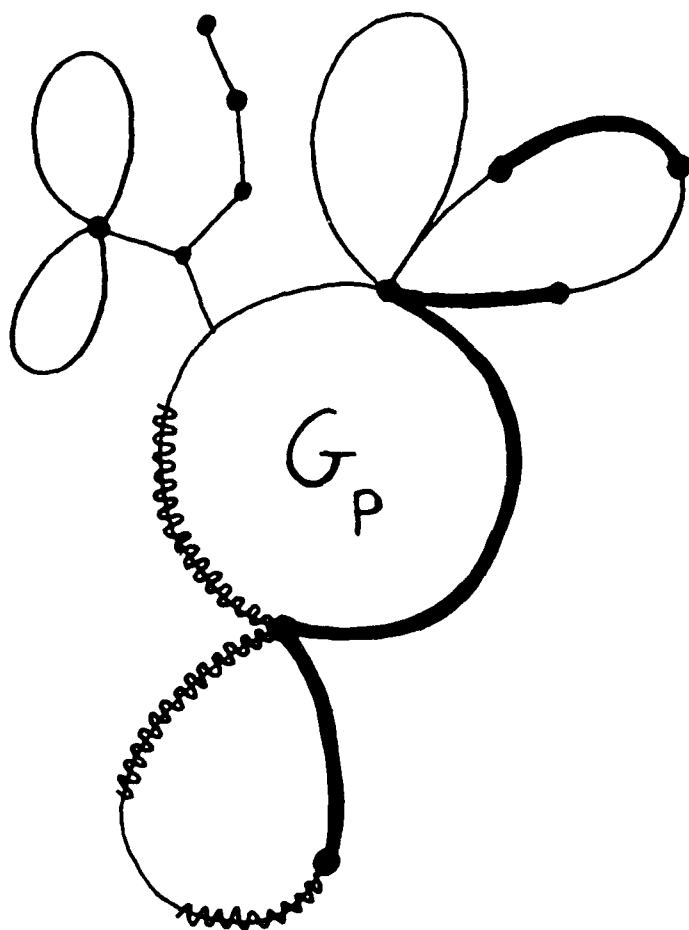


Figure 3.6: Possible trace intersections for different homotopy classes.

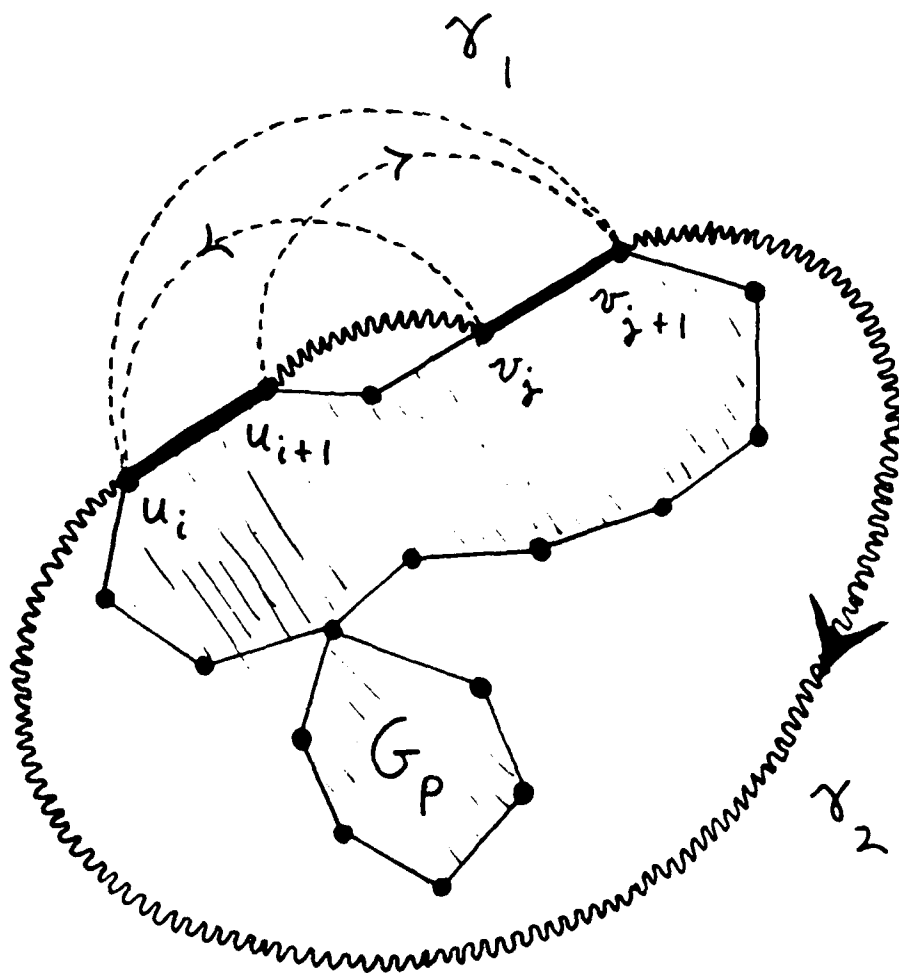


Figure 3.7: Construction for Lemma 3.2.3.

$$\gamma_1 = (u_i, u_{i+1}), (u_{i+1}, v_{j+1}), (v_{j+1}, v_j), (v_j, u_i)$$

is a simple closed curve on the surface S . Since (u_i, u_{i+1}, v_{j+1}) is a face, and (u_i, v_j, v_{j+1}) is a face, the interior of γ_1 is homeomorphic to an open disk. Draw another simple closed curve γ_2 on S with the following properties: (1) the interior of γ_2 is homeomorphic to an open disk and disjoint from the interior of γ_1 ; (2) γ_2 shares the edges $(u_i, u_{i+1}), (v_j, v_{j+1}) \in G_P$ with γ_1 , but no other points of γ_1 ; (3) the remainder of G_P is contained in the interior of γ_2 . If we traverse γ_2 on S clockwise, we visit the vertices $u_i, u_{i+1}, v_j, v_{j+1}$ in precisely this order. Since the interior of γ_1 is disjoint from the interior of γ_2 , if we traverse γ_1 on S counterclockwise we visit these vertices also in the same order. But this is a contradiction, since (u_{i+1}, v_j) is not an edge that belongs to γ_1 . ■

The next lemma relates the number of homotopy classes to the genus of the surface.

Lemma 3.2.4. [HI] If the surface S has genus $g \geq 1$, then any planar-nonplanar decomposition of G has at most $6g - 3$ homotopy classes.

Proof. Fix a planar-nonplanar decomposition (R, G_P, \mathcal{E}_N) of G . Then $\mathcal{E}_N \neq \emptyset$; otherwise G is planar. Draw a circle around the planar embedding of G_P that intersects each nonplanar edge exactly twice. (See Figure 3.8.) Create a new graph H by placing a new vertex at each such intersection, and erasing everything from the interior of the circle. H now has a 2-cell embedding in S and a planar-nonplanar decomposition with the same number of homotopy classes as G . The circle is the planar part of H , and the remaining edges are the nonplanar part. Let F be a face of H determined by two homotopic edges of H . Notice F is bounded by a 4-cycle consisting of two nonplanar edges, and two planar edges. If we contract the two planar edges on the boundary of F and eliminate the multiple edge, the resulting graph H' has a 2-cell embedding in S and a decomposition with the same number of homotopy classes as G . By repeating this procedure, we eventually end up with

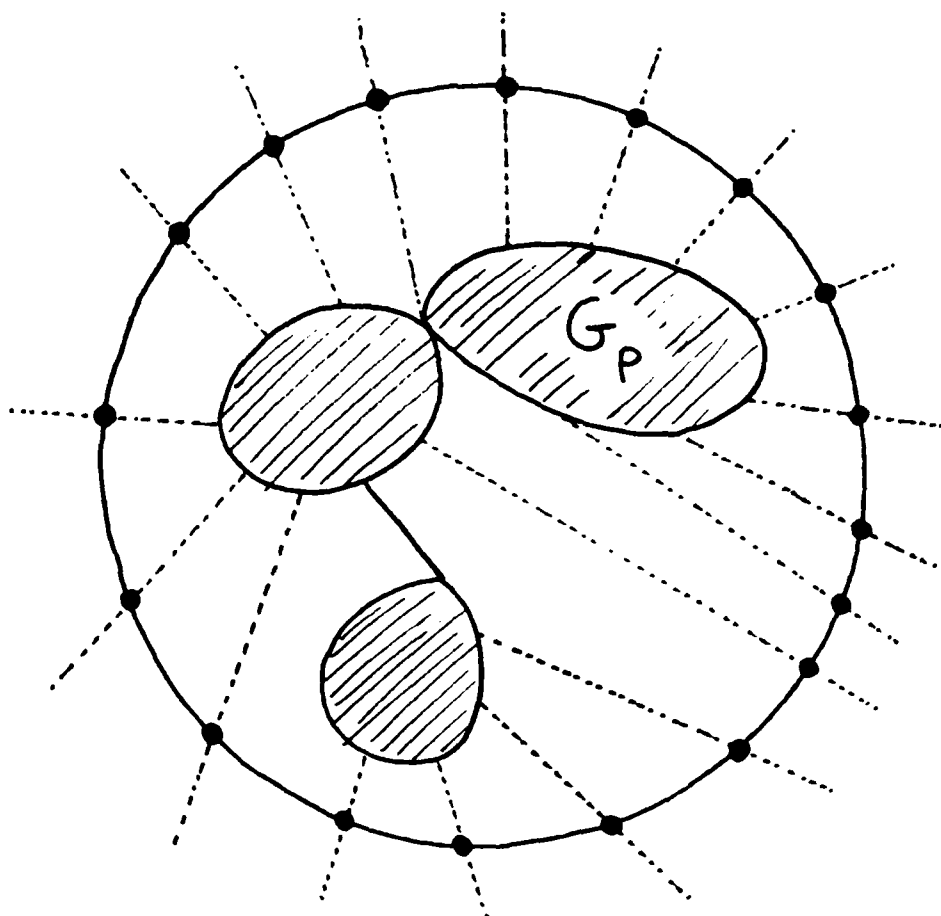


Figure 3.8: Construction for Lemma 3.2.4.

a graph H' that has the same number of homotopy classes as G , and such that each homotopy class consists of a single edge.

Consider the embedding of $H' = (V', \mathcal{E}')$ on S , and let $v = |V'|$, $e = |\mathcal{E}'|$, $h = \#$ of homotopy classes, and $f = \#$ of faces. Euler's formula for orientable surfaces of genus g is

$$v - e + f = 2 - 2g.$$

Notice that $2e = 3v$ as H' is regular of degree 3, and $2h = v$ since each vertex is incident to exactly one homotopy class. The interior face of H' has exactly v incident edges, and all of the remaining $f - 1$ faces have at least 6 incident edges (if a face had only 4 incident edges, the two nonplanar edges bounding the face would have been merged in the contracting steps.) Thus $6(f - 1) + v \leq 2e$. Combining this last inequality with the other equations yields $h = v/2 \leq 6g - 3$. ■

3.2.3 The Book-Embedding Algorithm

There are two stages to the algorithm BOOK-EMBED, which book-embeds a connected graph initially drawn on an orientable surface without crossings. The first stage, DECOMPOSE (see Figure 3.9), takes the input graph G which is 2-cell embedded in an orientable surface S of genus g , and outputs a planar-nonplanar decomposition of G . The second stage, PAGES, takes as input a planar-nonplanar decomposition of G and outputs a book-embedding of G using $O(g)$ pages.

The first step of DECOMPOSE extends G to a surface triangulation $G_T = (V_T, \mathcal{E}_T)$ without loops or multiple edges. This is done as follows. Consider a non-triangular face F in the embedding of G . Place a vertex v_f inside F and triangulate F by adding edges that connect v_f to all vertices on the boundary of F . Since the boundary of F may have multiple vertices, it is possible that this triangulation introduces multiple edges. Suppose (v_f, v) is one such copy of a multiple edge. Then (v_f, v) is incident to two triangular faces (v_1, v_f, v) and (v_2, v_f, v) where v_1 and v_2 are on the boundary of F . Notice $v_1 = v_2$, otherwise G would have multiple edges.

```

(1)   $G_T = (V, E_T) \leftarrow$  a surface triangulation of  $G$ 
(2)   $G_P = (V_P, E_P) \leftarrow$  some face of  $G_T$ 
(3)  while  $V_P \neq V$  do
(4)      if  $\exists$  safe vertex  $v_k$  (with respect to  $v_i \rightarrow v_j$ )
(5)          then (* add a safe vertex *)
(6)               $V_P \leftarrow V_P \cup \{v_k\}$ 
(7)               $E_P \leftarrow E_P \cup \{(v_i, v_k), (v_j, v_k)\}$ 
(8)          else (* start a new block *)
(9)               $w' \leftarrow$  newest vertex in  $V_P$  incident to a vertex in  $V - V_P$ 
(10)              $w \leftarrow$  vertex in  $V - V_P$  adjacent to  $w'$  (* see text *)
(11)              $V_P \leftarrow V_P \cup \{w\}$ 
(12)              $E_P \leftarrow E_P \cup \{(w, w')\}$ 
(13)         while  $\exists$  safe edge  $(v_i, v_k) \in E_N$  do
(14)              $E_P \leftarrow E_P \cup \{(v_i, v_k)\}$  (* add a safe edge *)
(15)         enddo
(16)     enddo

```

Figure 3.9: Algorithm DECOMPOSE (taken from [HI].)

Add a vertex v_s to the middle of the edge (v_f, v) , and triangulate the two incident faces by adding the edges $(v_1, v_s), (v_2, v_s)$. The multiple edge (v_f, v) has now been eliminated. If we perform this entire procedure on all nontriangular faces and all multiple edges introduced, we end up with the desired triangulation G_T .

After triangulating, DECOMPOSE is ready to generate a planar-nonplanar decomposition of G_T . Initially DECOMPOSE chooses one triangular face as the planar part, and adds faces incrementally. At any stage of the algorithm, $G_P = (V_P, \mathcal{E}_P)$ represents the planar part constructed so far, and $\mathcal{E}_N = \mathcal{E}_T - \mathcal{E}_P$ represents those edges of G_T which are currently nonplanar. There are two kinds of edges in \mathcal{E}_N . Those with both endpoints in the current V_P will remain nonplanar to the end of DECOMPOSE. But those with at least one vertex in $V - V_P$, still have a chance of becoming edges of G_P at a later time. Fix a clockwise orientation for the current boundary of G_P . If $v_i - v_j - v_k$ is a trace of G_P and (v_i, v_j, v_k) is a *face* of the embedding, then $(v_i, v_k) \in \mathcal{E}_T$ is called a **safe edge** (See Figure 3.10). If $v_i - v_j$ is a trace of G_P , $v_k \in V - V_P$, and (v_i, v_j, v_k) is a *face* of the embedding, then v_k is a **safe vertex** with respect to $v_i - v_j$ (See Figure 3.11). Clearly, in a planar-nonplanar decomposition of G , there can be no safe vertices (since $V - V_P = \emptyset$), and no safe edges (otherwise G_P is not maximal with respect to properties (1) and (2) in the definition of planar-nonplanar decomposition). Typically, DECOMPOSE chooses safe vertices (steps (7-8)) and safe edges (steps (14-16)) and adds them to G_P . When it happens that we cannot extend G_P in this manner, an unsafe vertex in $V - V_P$ adjacent to the boundary of G_P is selected in (steps (9-10)).

As the algorithm progresses, vertices and edges of G_P are assigned "ages" to indicate the relative order in which they were added. Steps (9-10) select a vertex $w \in V - V_P$ that is adjacent to the newest vertex w' on the boundary of G_P . The choice of w is as follows. Let (x, w') be any edge G_P that is incident to w' . Starting with the edge (x, w') , sweep clockwise about w' on the surface S until the first edge that connects w' to a vertex in $V - V_P$ is encountered. Take w to be this vertex.

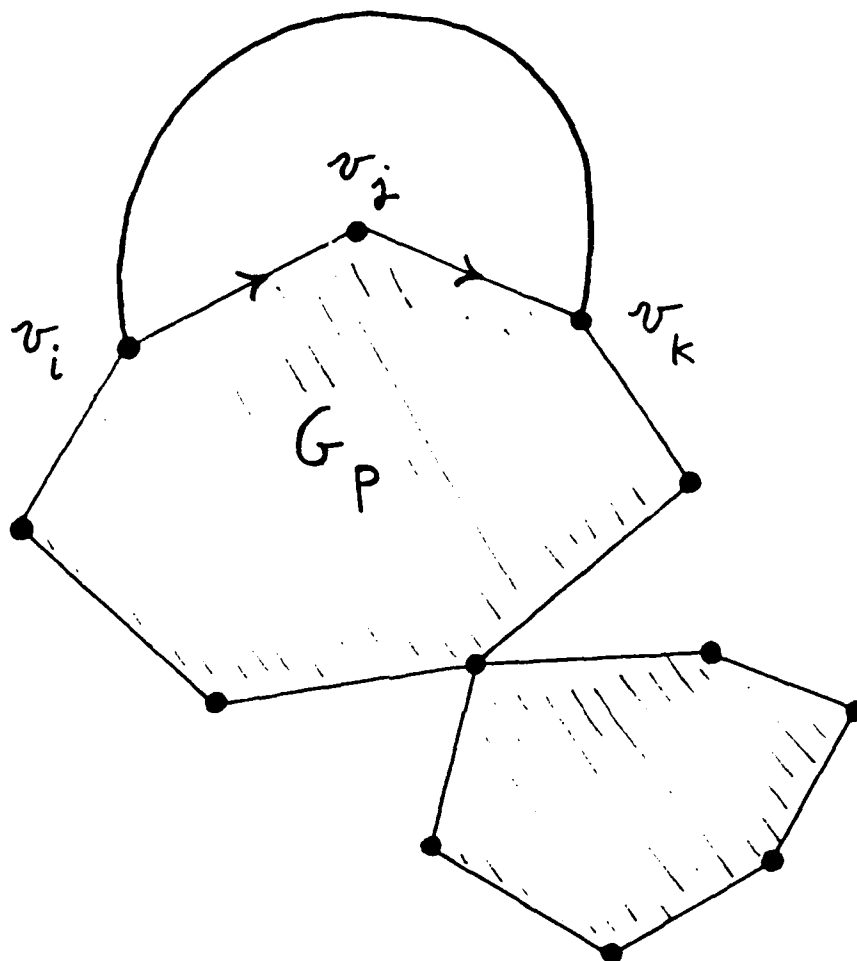


Figure 3.10: A safe edge.

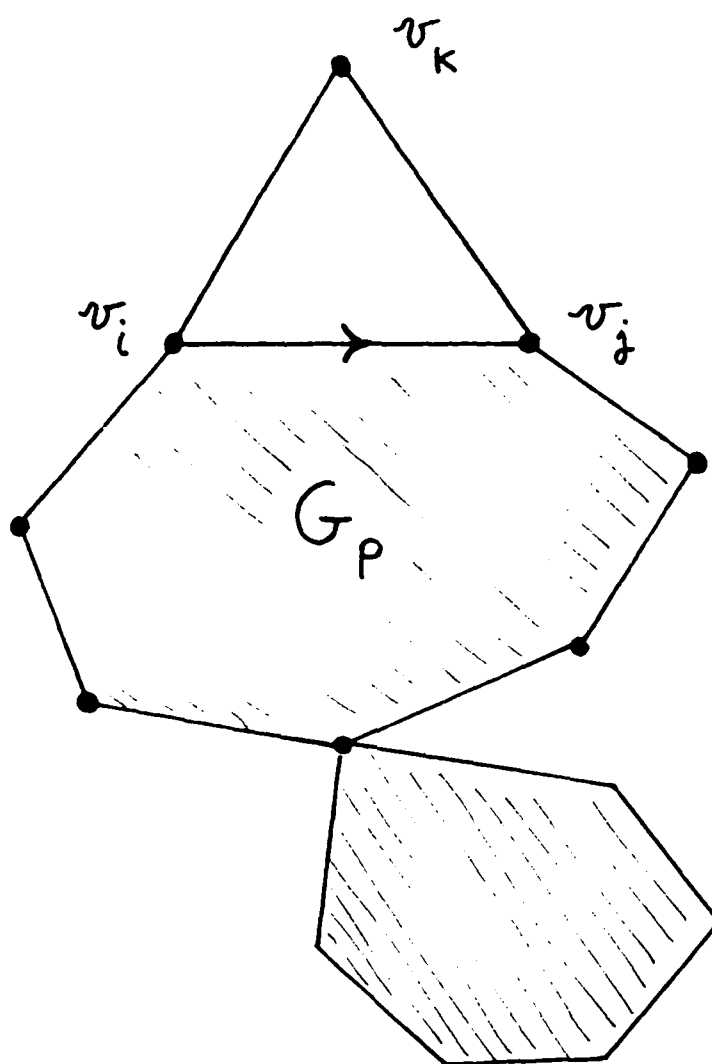


Figure 3.11: A safe vertex.

(See Figure 3.12.)

Whenever steps (9-12) are executed, the current block of G_P cannot be extended. Hence the algorithm initiates a new block of G_P . By using the newest w' in step (9), the blocks of G_P are constructed in depth-first order—the new block and all its descendants will be completed before the current block is again visited.

At any stage of DECOMPOSE homotopy classes are defined on the nonplanar edges with both endpoints in the current V_P . With the addition of new vertices in steps (6-7) and (9-12), and new edges in step (14), homotopy classes are originating, growing, and merging. To avoid confusion, we want to distinguish between homotopy classes that exist while DECOMPOSE is still running, and homotopy classes that exist at the completion of DECOMPOSE. Call the former classes **transitional** and the latter classes **conclusive**.

Lemma 3.2.5. [HI] Let C be a conclusive homotopy class with traces $T_1 = u_1, u_2, \dots, u_s$ and $T_2 = v_1, v_2, \dots, v_t$. Then C has one of three forms:

1. *No edge of C has both endpoints on the same block of G_P . In this case, at least one trace of C is **degenerate** (i.e. is a single point.)*
2. *Some edge of C has both endpoints on the same block of G_P . In this case one of two situations hold:*
 - (a) *There is a subtrace $S_2 = v_i, v_{i+1}, \dots, v_j$ of T_2 such that both T_1 and S_2 have all their vertices in the same block of G_P and every edge of C incident to v_1, \dots, v_i is incident to the endpoint u_1 and every edge of C incident to v_j, \dots, v_t is incident to the endpoint u_s .*
 - (b) *There is an initial subtrace $S_1 = u_1, \dots, u_i$ of T_1 and an end subtrace $S_2 = v_j, \dots, v_t$ of T_2 such that both S_1 and S_2 have all their vertices in the same block of G_P and every edge of C incident to v_1, \dots, v_j is incident to the endpoint u_1 , and every edge of C incident to u_1, \dots, u_i is incident to the endpoint v_t .*

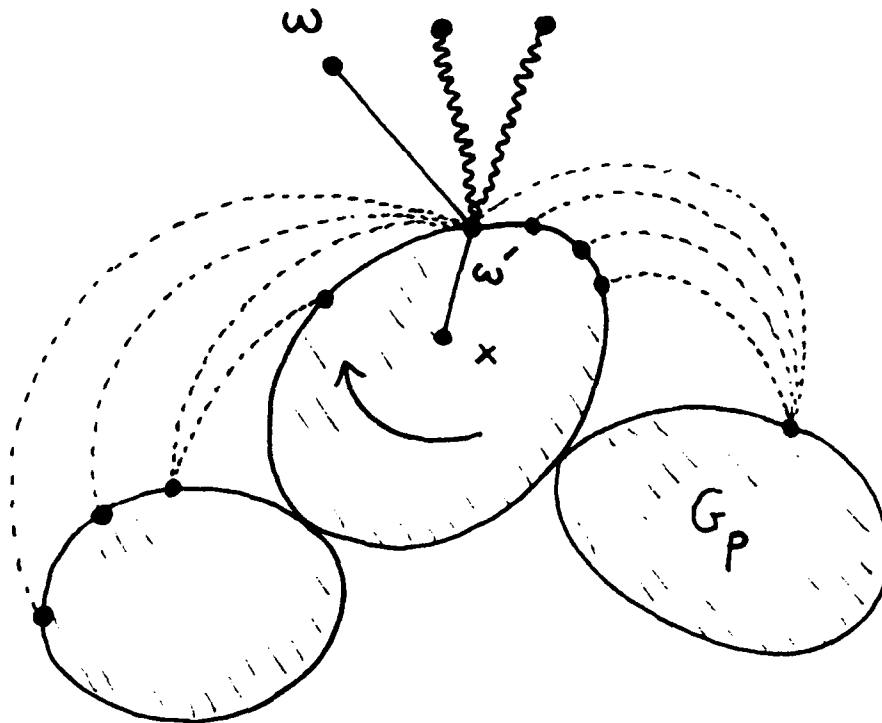


Figure 3.12: Choice of w in step (10) of DECOMPOSE.

Furthermore, neither T_1 nor T_2 is self-intersecting, and T_1 does not intersect T_2 .

Proof. There are two situations to consider.

- (i) *Suppose some first edge of C connects a vertex in the current block to a vertex in a different block.* We assume this edge is introduced by execution of steps (6-7); the case where steps (9-12) introduce some first edge is similar.

Suppose (v_k, z) is some first edge of C , and z is not in the current block. Then no other edge incident to v_k is homotopic to (v_k, z) ; otherwise, v_k is a safe vertex for the block z is in and would have been added earlier to that block. If v is any future vertex added to G_P and $(v, z) \in C$, then v cannot introduce an edge homotopic to (v, z) for the same reason. Thus C consists of edges incident only to z .

Let $T_1 = z$ and T_2 be the other trace of C . We claim T_2 cannot be self-intersecting (at an articulation point α of G_P). For suppose $u_i \rightarrow \dots \rightarrow u_j$ is a subtrace of T_2 that lies entirely on a "leaf" block B of G_P (viewing the block structure of G_P as a tree), and $\alpha = u_i = u_j$ is an articulation point of G_P . Then (α, u_{i+1}, z) and (α, u_{j-1}, z) are faces of the surface embedding, which means that B contains all the planar edges incident to α (remember (α, z) , (u_{i+1}, z) and (u_{j-1}, z) are nonplanar edges). But this contradicts the assumption that α is an articulation point of G_P .

- (ii) *Suppose (v_k, z) is a first edge of C and v_k is in the same block B as z .* Let $S_1 = x_1 \rightarrow x_2 \rightarrow \dots \rightarrow x_s$ and $S_2 = y_1 \rightarrow y_2 \rightarrow \dots \rightarrow y_t$ be the traces of the transitional homotopy class that contains (v_k, z) just before step |B|(9) is executed. Both S_1 and S_2 lie entirely on B since the *first* edge of C has both endpoints in B and only the block B is extended before execution of step (9). Notice $x_1 \neq y_1$ and $x_s \neq y_t$ as G does not have multiple edges. Therefore S_1 and S_2 do not intersect.

Clearly, the only vertices of these traces that can be adjacent to vertices in $V - V_F$ are the endpoints. If adding w in step (10) is going to extend this transitional homotopy class, either $(x_1, w), (y_1, w) \in \mathcal{E}$ or $(x_s, w), (y_t, w) \in \mathcal{E}$. Without loss of generality, assume the second possibility holds. Also without loss of generality, assume y_t is newer than x_s . Then in step (10), we have $w' = y_t$, and the addition of w extends the trace S_2 to $S_2 \rightarrow y_{t+1}$, where $w = y_{t+1}$. Let B' be the new block initiated by the addition of y_{t+1} . As long as we can add vertices to B' and its descendants, the trace S_1 is not extended (otherwise there would be safe vertices for B). Let $T_2 = S_2 \rightarrow y_{t+1} \rightarrow \dots \rightarrow y_{t+q}$ be the extended trace at the time we complete B' and all its descendants. All the nonplanar edges of C contributed by y_{t+1}, \dots, y_{t+q} are incident to x_s (otherwise there would be safe vertices for B), and the extended trace T_2 is never self-intersecting (by the same argument used in case (i).) Thus y_{t+q} is not in B . We claim the trace S_1 can never be extended beyond x_s . For contradiction, suppose we could extend S_1 to $S_1 \rightarrow x_{s+1}$. Then $(x_s, x_{s+1}) \in \mathcal{E}_F$, and $(x_{s+1}, y_{t+q}) \in \mathcal{E}_V$ is homotopic to $(x_s, y_{t+q}) \in \mathcal{E}_N$. But before the addition of x_{s+1} to G_F , the vertex y_{t+q} , which is newer than x_s , is adjacent to $x_{s+1} \in V - V_F$. Hence a new block should have been started at y_{t+q} , contradicting our assumption that B' and its descendants were completed. Thus S_1 cannot be extended beyond x_s . Similar arguments applied to the other sides of S_1 and S_2 give traces of the desired form. ■

Figures 3.13, 3.14, and 3.15 show homotopy classes of types 1, 2(a), and 2(b), respectively.

The second stage, PAGES, takes as input a planar-nonplanar decomposition of G , and outputs a book embedding of G with $O(g)$ pages. This is done as follows. By an algorithm of Yannakakis [Ya], any biconnected planar graph can be given a 4-page embedding in linear time which maintains the cyclic order of the boundary vertices. For an arbitrary planar graph, an algorithm of Heath [He] combines the

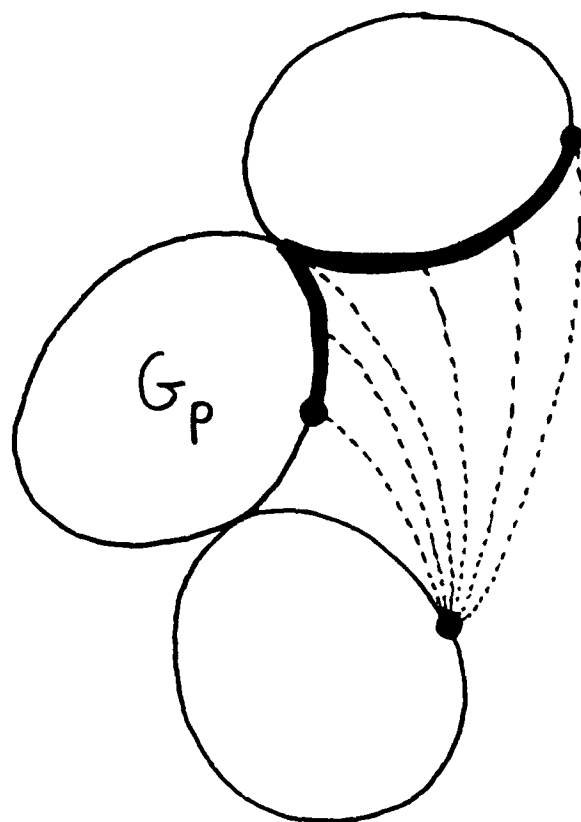


Figure 3.13: Homotopy class of type 1.

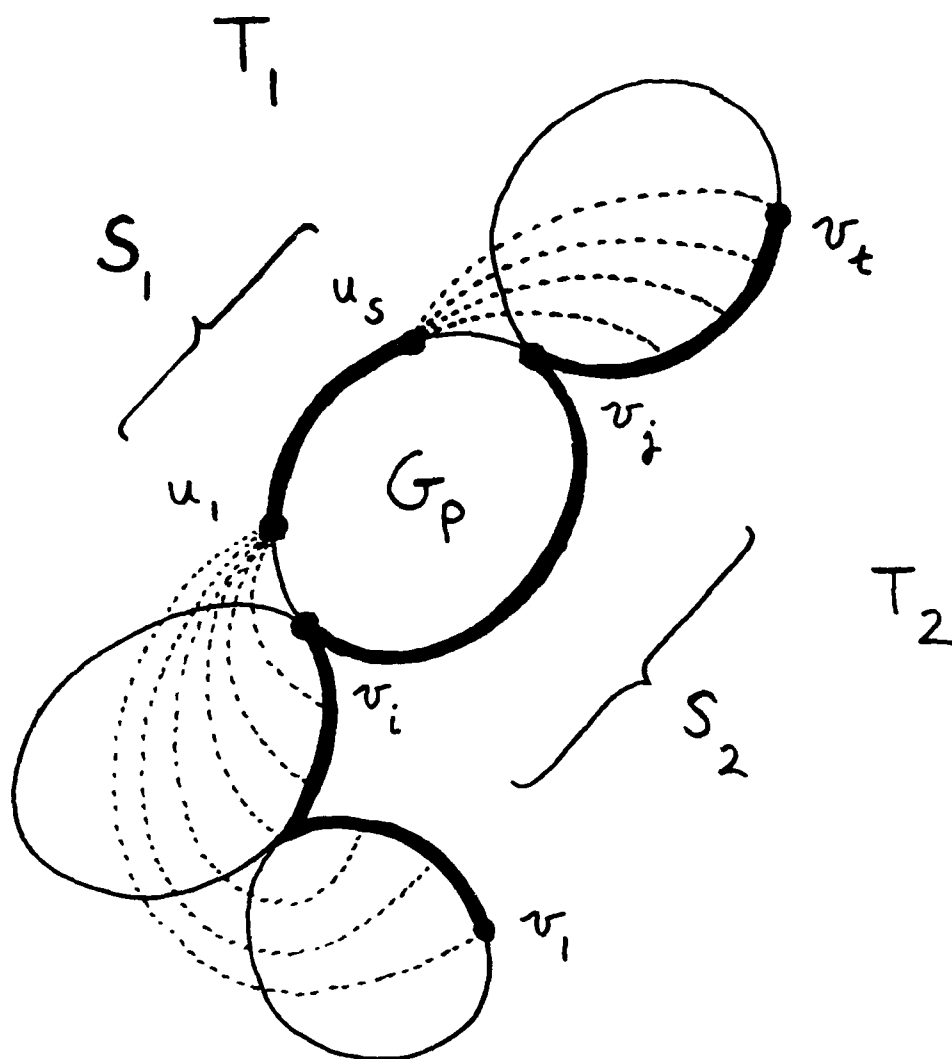


Figure 3.14: Homotopy class of type 2(a).

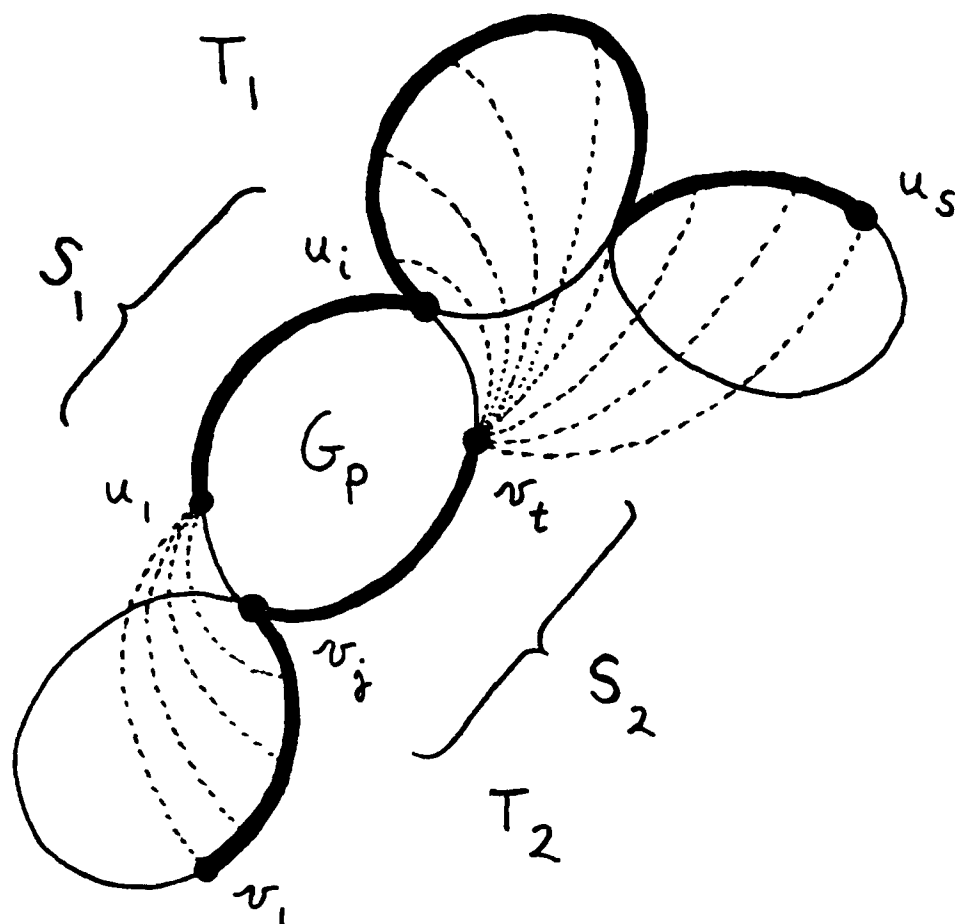


Figure 3.15: Homotopy class of type 2(b).

4-page embeddings for all the blocks in linear time, and yields a 4-page embedding for the entire graph that maintains the cyclic order of the boundary vertices on each block. Hence G_P has such a 4-page embedding. A vertex order has now been established, and all the edges of \mathcal{E}_P have been assigned to pages. Now we allocate 3 pages for each homotopy class in \mathcal{E}_N . (In the degenerate case, one page suffices for the homotopy class.) Allot one page for the edges between vertices of the same block, and two pages for the remaining edges of the homotopy class (the two "fans"). The resulting embedding requires at most $4 + 3(6g - 3) = 18g - 5$ pages.

The entire algorithm BOOK-EMBED can be shown to run in time $O(n + g)$. (See [HI].)

3.2.4 Some Additional Definitions

Certain vertices belonging to the traces of a homotopy class will have special significance in the sections that follow. These are called **critical vertices**. Suppose C is a type 2 ((a) or (b)) homotopy class with traces T_1 and T_2 . Let S_1 and S_2 be the *longest* subtraces of T_1 and T_2 respectively, such that S_1 and S_2 lie entirely on the same block. The critical vertices of C are the endpoints of S_1 and S_2 . For example, u_1, u_i, v_j , and v_l are the critical vertices of the homotopy class in Figure 3.15. The critical vertices of a type 1 homotopy class are simply its degenerate traces. The important fact about critical vertices is that they cover all nonplanar edges incident to articulation points of G_P .

Remove the critical vertices from S_1 and S_2 , and denote the resulting subtraces by S'_1 and S'_2 , respectively. We will refer to these as the **foundational subtraces** of T_1 and T_2 , respectively.

3.3 The Chain Graph Lemma

A **chain graph** consists of a set V of vertices, a partition $\{C_1, C_2, \dots, C_n\}$ of V into linearly ordered subsets called **chains**, and a collection \mathcal{E} of two-element subsets of V (called **edges**), each of which is incident to two *distinct* chains.

An example of a chain graph is shown in Figure 3.16.

Chain graphs are important objects in our embedding strategy for genus g graphs. The basic approach we take in Sections 3.4 and 3.5 is as follows. We start initially with an $O(g)$ -page book embedding B of $G = (V, \mathcal{E})$ given by the Heath-Trail algorithm. Considering the embedding B as it stands, homotopy classes may interact in such a way as to require $\Omega(g)$ pages with respect to the given vertex ordering. Our goal will be to permute the vertices in a manner that keeps homotopy classes intact, but uncrosses them as much as possible without disturbing the edges of G_P too badly. This simple intuition is made more precise below.

The embedding B dictates a decomposition of G into a constant number of subgraphs, and a partition of the linear ordering of V into intervals. We introduce an equivalence relation on the edges of each subgraph by saying, roughly, that two edges are in the same equivalence class iff they connect the same pair of intervals. Equivalence classes behave and interact in a nice manner that allows us to represent each subgraph by a “quotient” chain graph—one chain for each interval and one edge for each equivalence class. Rearranging the chains of one of these chain graphs corresponds to rearranging the intervals of V in exactly the same way. The representation of subgraphs by quotient chain graphs satisfies three very nice properties: (a) no two edges of a quotient chain graph connect the same pair of chains; (b) each quotient chain graph has only $O(g)$ edges, and (c) any “book embedding” of a quotient chain graph in t pages corresponds to a book embedding of the represented subgraph in t pages. To finish our construction, we apply a lemma that says any chain graph with E edges, no two of which are incident to the same pair of chains, can be embedded in $O(\sqrt{E})$ pages with respect to almost any linear juxtaposition

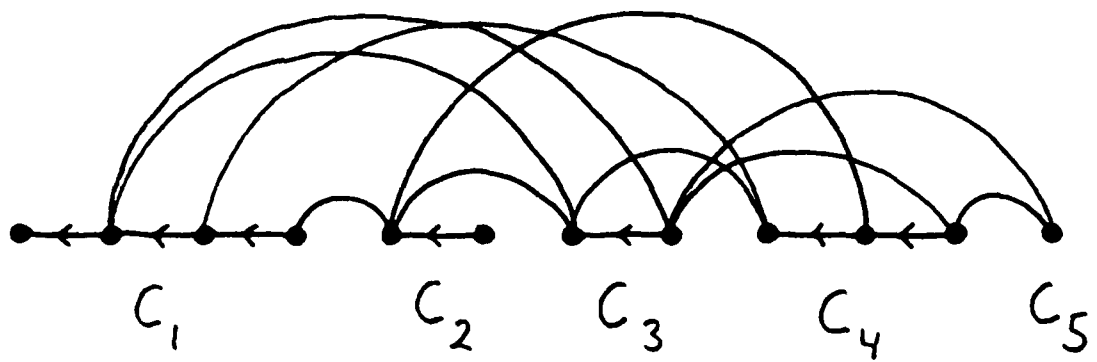


Figure 3.16: A chain graph.

of its chains. Ultimately, this allows us to rearrange the intervals of V to obtain a new linear ordering for which we can simultaneously embed each subgraph of G in $O(\sqrt{g})$ pages.

The outline above contains slight inaccuracies, but does indicate the basic approach. In actuality, we need a slightly stronger form of the last lemma about embedding chain graphs and that is what we develop now.

Let H be a chain graph. A **chain subgraph** of H is a chain graph whose chain set and edge set are subsets of H 's chain set and edge set, respectively.

H is **star-linked** if the bipartite graph induced by any pair of chains is a $K_{1,r}$ for some $r \geq 0$. In the case that $r = 0$ or 1 for every pair of chains, then H is **1-linked**.

A **linear ordering** or **permutation** of the chains in H is a linear arrangement of the vertices in which chains do not interleave and the linear order within each chain is preserved. It is useful to think of a linear ordering as being *horizontal*, so smaller elements are to the left and larger elements to the right.

We can now state our main lemma precisely.

Chain Graph Lemma. Let H be a star-linked chain graph with E edges on n chains, where $E \geq n$. Given a random linear ordering π of the chains, the probability that H can be embedded in $28\sqrt{E}$ pages with respect to π , is at least $1 - (1/2)^{\sqrt{E}}$.

Again, suppose H is a chain graph. By collapsing each chain to a single point and identifying multiple edges, we obtain the **quotient graph**, $Q(H)$.

Proving the Chain Graph Lemma is not simply a matter of applying Theorem 2.2.1 to the quotient graph $Q(H)$. This is because a page of $Q(H)$ may correspond to many pages of H . Although our proof of the Chain Graph Lemma will essentially mimick the proof of Theorem 2.2.1, there is more subtlety involved. (We will explain why shortly.) To get started, we need a few definitions and preliminary lemmas analogous to those of Chapter 2.

A **bipartite chain graph** (BCG) is a chain graph for which it is possible to assign each chain one of two colors, say LEFT or RIGHT, so that no edge connects two chains of the same color. A **2-colored bipartite chain graph** (2BCG) is a BCG with an associated 2-coloring of the chains such that no edge connects two chains of the same color. So, for example, if H is a BCG and $Q(H)$ has exactly d connected components, then H can be 2-colored in 2^d different ways with the colors LEFT and RIGHT.

A 2BCG is **canonically ordered** if its chains are linearly ordered so that all LEFT chains precede all RIGHT chains.

Given a (horizontal) linear ordering of the chains in a chain graph and an edge e , denote the left and right endpoints of e by $l(e)$ and $r(e)$ respectively.

A 2BCG is **completely crossing** with respect to the canonical ordering π if its edges can be labeled e_1, e_2, \dots, e_k so that

$$l(e_1) <_{\pi} l(e_2) <_{\pi} \dots <_{\pi} l(e_k) \text{ and } r(e_1) <_{\pi} r(e_2) <_{\pi} \dots <_{\pi} r(e_k).$$

Intuitively, a 2BCG is completely crossing if no edge can be placed on the same page as another edge with respect to the vertex ordering π . If a star-linked 2BCG is completely crossing, then clearly it must be 1-linked.

We have the obvious analog of Lemma 2.2.1 for chain graphs.

Lemma 3.3.1. Let π canonically order the 2BCG H . If at most k edges are completely crossing with respect to π , then H can be embedded in at most k pages with respect to π .

We also have an analog of Lemma 2.2.2.

Lemma 3.3.2. A star-linked chain graph H on n chains can be embedded in $n + 1$ pages with respect to any linear ordering of the chains.

Proof. By the proof of Lemma 2.2.2, K_n can be embedded in $\lceil n/2 \rceil$ pages so that the edges on any page form a subgraph of a *path*. We apply this as follows. Obtain

the quotient graph $Q(H)$ and fix a linear ordering π of its vertices. Since $Q(H)$ is a subgraph of K_n , it can be embedded in $\lceil n/2 \rceil$ pages with respect to π , and the edges on any page can be colored with two colors so that adjacent edges are not assigned the same color. To embed H in $n + 1$ pages with the corresponding order on chains, take those edges of $Q(H)$ that are on the same page and of one color, and embed their preimages from H on a single page. We thus allot two pages to H , for every one page of $Q(H)$. ■

As mentioned earlier, the Chain Graph Lemma is rather more difficult to establish than Theorem 2.2.1, and the reason is that we know nothing at the moment about the structure of completely crossing 2BCG's. In particular, we do not know how many chains belong to a completely crossing 2BCG with k edges. Thus we do not know how many canonical orderings it has, or how many of them make it completely crossing. In Theorem 2.2.1, we were relying heavily on the fact that a completely crossing 2BG is a matching. This makes it easy to obtain the exact probability that a 2BG is completely crossing with respect to a random canonical ordering. To get even a good upper bound on the analogous probability for a 2BCG, we need more information regarding the structure of completely crossing 2BCG's.

In the four lemmas that follow, J is a 1-linked 2BCG and π a canonical ordering of J .

Lemma 3.3.3. If J is completely crossing with respect to π , then $Q(J)$ is a forest.

Proof: If $Q(J)$ is not a forest, then it contains an even cycle. Let γ be a 1-linked 2BCG such that $Q(\gamma)$ is a cycle with $2k$ edges. We aim to show that γ cannot be made completely crossing with respect to any canonical ordering. Enumerate the edges of γ in cycle order e_1, e_2, \dots, e_{2k} , so that e_{2k} shares its LEFT chain with e_1 and its RIGHT chain with e_{2k-1} . Now suppose e_1, \dots, e_{2k-1} is completely crossing with respect to a canonical ordering π and that without loss of generality, e_1 's left

endpoint is leftmost of all edges. A simple induction on i demonstrates that

$$l(e_1) <_{\pi} l(e_2) <_{\pi} \cdots <_{\pi} l(e_i) \text{ and } r(e_1) <_{\pi} r(e_2) <_{\pi} \cdots <_{\pi} r(e_i)$$

for all $i \leq 2k - 1$. In particular, this holds for $i = 2k - 1$. But we know that e_{2k} shares its LEFT chain with e_1 and its RIGHT chain with e_{2k-1} , and hence can be placed on the same page as e_2 . ■

Lemma 3.3.4. If $Q(J)$ is a tree, then at most one canonical ordering makes J completely crossing.

Proof: Do a breadth-first search of J from some initial starting chain. A trivial induction on i shows that the structure of J forces the canonical ordering of chains that makes edges in levels $1, \dots, i$ completely crossing (if such an ordering exists.)

■

Lemma 3.3.5. Let J be completely crossing with respect to π and suppose $Q(J)$ is the union of a tree T and an isolated edge e . Then with respect to π , one chain of e succeeds all LEFT chains of T and precedes all RIGHT chains of T , and the other chain either precedes or succeeds all chains of T .

Proof: Suppose, for contradiction, that the chains of e are not positioned as stated. Denote the LEFT chain of e by $L(e)$, and the RIGHT chain of e by $R(e)$. Let J' be the component of J whose quotient graph is T , and let $\mathcal{C}(J')$ be the set of chains in J' . We define four subsets of $\mathcal{C}(J')$ as follows. Let L_{in} be the set of LEFT chains between $L(e)$ and $R(e)$; let R_{in} be the set of RIGHT chains between $L(e)$ and $R(e)$; let L_{out} be the set of LEFT chains not between $L(e)$ and $R(e)$; let R_{out} be the set of RIGHT chains not between $L(e)$ and $R(e)$. Since J is completely crossing, no edge of J' can be placed on the same page as e with respect to π . This means there can be no edge connecting chains in $L_{\text{out}} \cup R_{\text{in}}$ with chains in $L_{\text{in}} \cup R_{\text{out}}$. Since neither union is empty, $Q(J')$ is not connected, contradicting the fact that T is a tree. ■

Combining Lemmas 3.3.4 and 3.3.5, we have

Lemma 3.3.6. If J is completely crossing with respect to π and $Q(J)$ is a forest with d trees, then exactly $d!$ canonical orderings make J completely crossing.

Notice that if $Q(J)$ is a forest with d trees, then J has $E(J) + d$ chains and hence can be canonically ordered in

$$(\# \text{ of LEFT chains})! \times (\# \text{ of RIGHT chains})! \geq \left(\frac{E(J) + d}{2} \right)!^2$$

different ways.

We are now ready to establish the Chain Graph Lemma. The proof from this point on is similar to that of Theorem 2.2.1.

Proof of Chain Graph Lemma. Pick a random linear ordering π of the chains of H . Partition the edges of H into $\log n$ "levels" as follows. To obtain the j -level edges ($1 \leq j \leq \log n$), first divide π into 2^j segments each with the same number of chains. Label the segments from left to right L.R.L.R.L. etc. Any edge that connects chains in an adjacent L.R pair of segments is a j -level edge. Notice that every edge of H is in a unique level.

Let A_k^j be the event that there exists a bipartite chain subgraph J of H with k edges, and a 2-coloring χ of J , such that J is canonically ordered, j -level, and completely crossing with respect to π . Since H is star-linked, such a J must be 1-linked. By Lemma 3.3.3, $Q(J)$ must also be a forest. We have

$$Pr[A_k^j] < \sum_{d=1}^k \underbrace{2^{j-1}}_1 \underbrace{\binom{E}{k}}_2 \underbrace{2^d \left(\frac{1}{2^j} \right)^{k+d}}_3 \underbrace{\frac{d!}{\left(\frac{k+d}{2} \right)!^2}}_4$$

where the sum is over the number of trees in $Q(J)$, and

- (1) is the number of adjacent L,R segment pairs at the j th level;
- (2) upper bounds the number of J in H (including a designated 2-coloring) such that $Q(J)$ is a forest with d trees;

- (3) upper bounds the probability that π canonically orders the chains of J by putting all LEFT chains in a fixed segment labeled L and all RIGHT chains in the adjacent segment labeled R;
- (4) upper bounds the probability that J is completely crossing with respect to π given that π canonically orders J .

Using the fact that

$$\left(\frac{k+d}{2}\right)! > \frac{(k+d)!}{2^{k+d}} > \frac{k!d!}{2^{k+d}} > \frac{k!d!}{2^{2k}}$$

we obtain

$$Pr[A_k^j] < 2^{2j-1} \binom{E}{k} 2^k \left(\frac{1}{2^j}\right)^{k+1} \frac{2^{2k}}{k!} < 2^{3k} \frac{E^k}{k!} \left(\frac{1}{2^j}\right)^k \frac{1}{k!} < \frac{(8e^2 E)^k}{(k \sqrt{2^j})^{2k}}.$$

If we let $k_j = 4e\sqrt{E}/\sqrt{2^j}$, then

$$Pr[A_{k_j}^j] < (1/2)^{4e\sqrt{E}/\sqrt{2^j}} < (1/2)^{4e\sqrt{n}/\sqrt{2^j}}$$

where the last inequality follows from the assumption $E \geq n$. The next step is to show that simultaneously we expect no more than $4e\sqrt{E}/\sqrt{2^1}$ completely crossing 1-level edges, no more than $4e\sqrt{E}/\sqrt{2^2}$ completely crossing 2-level edges, no more than $4e\sqrt{E}/\sqrt{2^3}$ completely crossing 3-level edges, and so on in a nice geometrically decreasing fashion. We have

$$\begin{aligned} & Pr[A_{k_1}^1 \vee A_{k_2}^2 \vee \dots \vee A_{k_{\log(n/\sqrt{E})}}^{\log(n/\sqrt{E})}] \\ & < Pr[A_{k_1}^1 \vee A_{k_2}^2 \vee \dots \vee A_{k_{\log \sqrt{n}}}^{\log \sqrt{n}}] \\ & < (1/2)^{4e\sqrt{n}/\sqrt{2^1}} + (1/2)^{4e\sqrt{n}/\sqrt{2^2}} + \dots + (1/2)^{4e\sqrt{n}/\sqrt{2^{\log \sqrt{n}}}} \\ & < (1/2)^{8\sqrt{n}}. \end{aligned}$$

Hence

$$Pr[\neg A_{k_1}^1 \wedge \neg A_{k_2}^2 \wedge \dots \wedge \neg A_{k_{\log(n/\sqrt{E})}}^{\log(n/\sqrt{E})}] > 1 - (1/2)^{8\sqrt{n}}.$$

Therefore, with probability at least $1 - (1/2)^{8\sqrt{n}}$, a random linear ordering of the chains allows the j -level edges to be embedded in $4e\sqrt{E}/\sqrt{2}^j$ pages (by Lemma 3.3.1) for all j in the range 1 through $\log(n/\sqrt{E})$. (We did not go all the way to level $\log n$ because as j approaches $\log n$, the quantity $Pr[A_k^j]$ approaches a constant, and thus we would not obtain a high probability result.)

Observe now that segments at level $\log(n/\sqrt{E})$ contain only \sqrt{E} chains. Hence the remaining edges, those in levels $1 + \log(n/\sqrt{E})$ and beyond, can be embedded in \sqrt{E} pages by Lemma 3.3.2. Thus, for "good" linear orderings, the total number of pages used to embed all the edges of H is at most

$$4e\sqrt{E}/\sqrt{2}^1 + 4e\sqrt{E}/\sqrt{2}^2 + \dots + 4e\sqrt{E}/\sqrt{2}^{\log(n/\sqrt{E})} + \sqrt{E} \leq 28\sqrt{E}. \blacksquare$$

3.4 The One-Block Case

In deriving the $O(\sqrt{g})$ upper bound on pagenumber for arbitrary genus g graphs, it is useful to focus first on a special case. For this section we will assume the Heath-Istrail algorithm yields a planar-nonplanar decomposition where G_P is a single biconnected component. This situation illustrates most of the ideas used in the general case, but is less complicated.

Let us suppose G_P consists of only one block. After applying the Heath-Istrail algorithm, the vertices of G are linearly ordered in a way that preserves the cyclic order of vertices on the boundary of G_P , the edges of G_P are embedded in 4 pages, and the $O(g)$ homotopy classes are assigned one to a page (there are no "fans.")

Call the above linear ordering on the vertices L . Let Δ be the partition of L into $O(g)$ intervals obtained by "cutting" L immediately before and after each critical vertex. Thus each critical vertex is placed in an interval by itself.

Divide the edges of G into 5 subgraphs, each on the vertex set V . Let G_0 be the graph consisting of all nonplanar edges, and let G_1, G_2, G_3, G_4 be the 4 pages of G_P .

For each $i \in \{0, \dots, 4\}$, define an equivalence relation \sim_Δ on the edges of G_i : say two edges are equivalent iff they connect the same pair of intervals. (We choose to ignore those edges of G_i with both endpoints in the same interval—all such edges will be placed on one page per subgraph at the end.)

The following properties hold for each G_i .

- Edges in the same equivalence class can be placed on the same page with respect to L . (Reason: For G_0 , this follows because each homotopy class is orientable (Lemma 3.2.3). For G_1, \dots, G_4 , this follows because each is a planar graph.)
- If e_1, e_2, e_3 are three edges incident to the same interval such that $l(e_1) <_L l(e_2) <_L l(e_3)$ and $e_1 \sim_\Delta e_3$, then $e_1 \sim_\Delta e_2 \sim_\Delta e_3$. In other words, an equivalence class incident to an interval determines a subinterval, no vertex of which is incident to another equivalence class. (Reason: The statement holds for G_0 because Lemma 3.2.2 says that traces from different homotopy classes do not share interior vertices since G_P consists of a single block. The statement holds for G_1, \dots, G_4 , because each is a planar graph.)

A typical G_0 and G_1 are illustrated in Figures 3.17(a) and 3.17(b), respectively.

Now we can represent each G_i by a **quotient chain graph** H_i in the following way. Represent each equivalence class of edges by a single edge, and each interval with t incident equivalence classes by a chain with t vertices. Make each chain vertex incident to exactly one edge, and connect the edges among chains in a manner that preserves the left-to-right order in which corresponding equivalence classes are incident to corresponding intervals.

For example, Figure 3.18 is the quotient chain graph of Figure 3.17(b).

Notice that each H_i has the same number of chains, one for each interval, and at most one edge connects a pair of chains in H_i . In general, chains in H_i and H_j

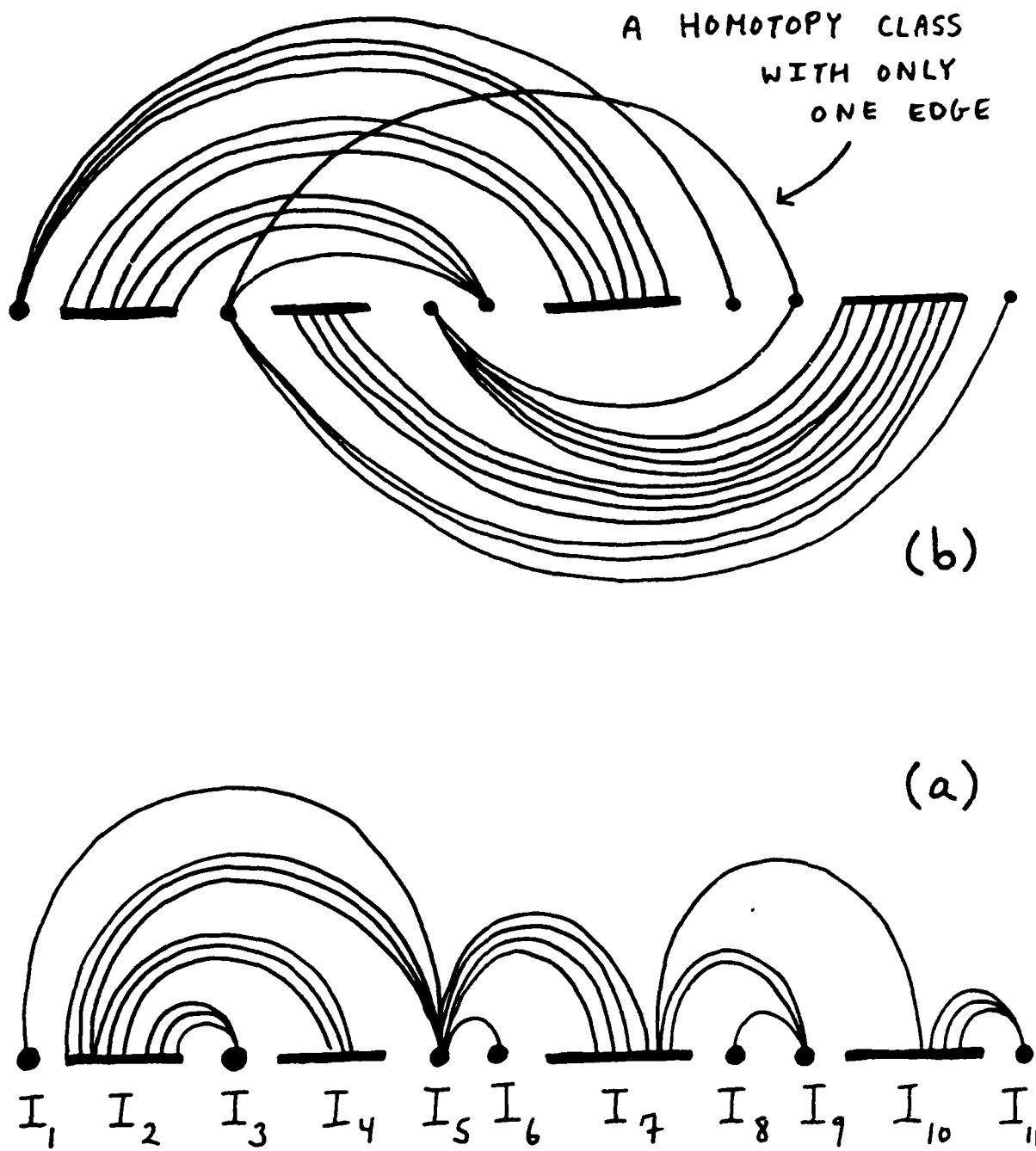


Figure 3.17: (a) A typical G_0 ; (b) A typical G_1 .

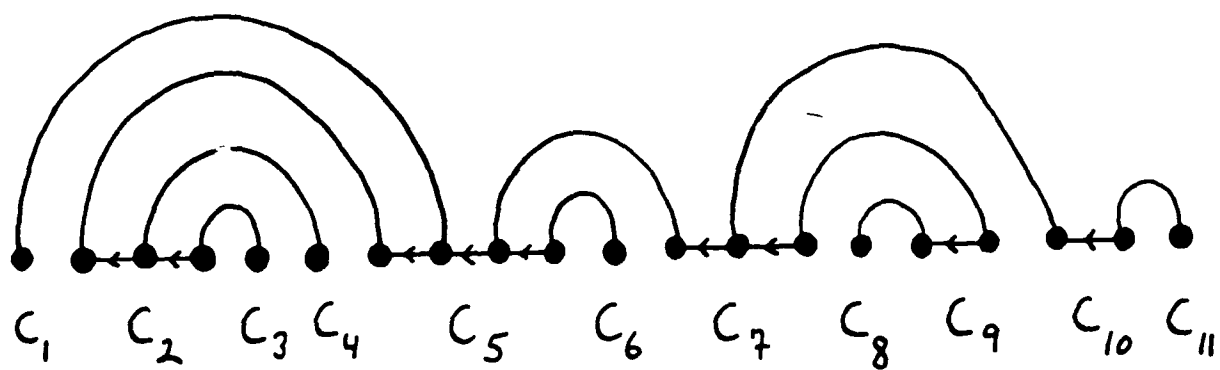


Figure 3.18: The quotient chain graph of Figure 3.17(b).

will be hooked up differently, and corresponding chains in each will not be the same length.

Lemma 3.5.1. Each H_i has only $O(g)$ edges.

Proof. For H_0 this follows because each homotopy class contributes at most 5 equivalence classes (here we are using Lemma 3.2.2 which says that traces from different homotopy classes do not share interior vertices since G_P consists of a single block). For $i \geq 1$, this follows because each H_i is an outerplanar chain graph on $O(g)$ chains. ■

If we imagine now permuting the intervals of L (i.e. linearly arranging the vertices of L so that intervals do not interleave and the linear order within each interval is preserved), this corresponds to permuting (in the same way) the order of the chains within each H_i . No matter what the permutation π , the following are true simultaneously for each G_i .

- Edges within an equivalence class can be placed on the same page.
- If two edges of H_i can be placed on the same page, then the corresponding pair of equivalence classes in G_i can be placed on the same page.

Therefore, if we can permute the intervals of L in such a way that each H_i can be simultaneously embedded in $O(\sqrt{g})$ pages with the corresponding order on chains, then we can embed all of G in $O(\sqrt{g})$ pages. But we know that such a permutation exists by the Chain Graph Lemma. ■

3.5 The General Case

Now that we can handle the situation where G_P has only one block, we are ready to attack the general case where G_P may have many blocks. First we need a little notation.

Let F be the sequence of vertices visited as we traverse the boundary of G_P in clockwise fashion from some initial starting vertex x_0 . Notice that articulation points have multiple occurrences in F . We refer to the first occurrence of an articulation point as the **primary** occurrence, and all later occurrences as **secondary**.

Let L_B be the linear ordering of the boundary vertices of G_P obtained by deleting all secondary occurrences of articulation points in F .

Let L be a linear ordering of all the vertices of G which allows G_P to be embedded in 4 pages and preserves the order of L_B . (Such an L exists by the algorithms of Yannakakis [Ya] and Heath [He2].)

The only significant difference between the general case and the one-block case is that now traces can intersect at interior vertices (necessarily articulation points by Lemma 3.2.2.) See Figure 3.6. Since articulation points are allowed only one occurrence in L_B , traces, in general, may not be segments of L_B . Thus traces may interleave in L_B .

Let G_A be the graph with vertex set V consisting of all edges of G that are incident to articulation points. The main purpose of this section is to construct a suitable partition Δ' of L , and a suitable equivalence relation on the edges of G_A , so that the following requirements hold: (1) edges in the same equivalence class can be placed on the same page with respect to L ; (2) every equivalence class of edges incident to an interval determines a subinterval no vertex of which is incident to another equivalence class; (3) only $O(g)$ intervals and $O(g)$ equivalence classes are present. With all these requirements met, we can obtain a well-defined quotient chain graph H_A having all the desired properties.

It is worth remarking that the partition Δ from the last section is not a suitable partition for the general case. This is because an articulation point that belongs to many blocks of G_P might be a critical vertex for a large number of homotopy classes, which means there could be far fewer critical vertices than there are homotopy classes. Hence we may not end up with enough intervals to allow us to uncross the

homotopy classes (requirements (1) and (2) may not be satisfied.)

Yet, the new partition Δ' is very similar in spirit to the partition Δ . The key properties we insist on are that it places each critical vertex in its own interval, and that it places each foundational subtrace in its own interval. (This gives us the freedom we need to uncross the homotopy classes.) The construction of Δ' requires a short sequence of steps. Let C be a homotopy class, T a trace of C , and v a critical vertex of C that belongs to T . If we view T as a segment of F , then by Lemma 3.2.2, T contains exactly one occurrence of v . Define a partition of F into $O(g)$ intervals as follows. For each triple C, T , and v as above, "cut" F immediately before and after the unique occurrence of v in T . (If v is an articulation point, this occurrence may be secondary in F . In this case, also "cut" F immediately before and after the primary occurrence of v in F .) Doing this for all such triples induces a partition of F into intervals. Restrict this partition of F to L_B (all secondary occurrences of articulation points vanish.) Notice that each critical vertex is placed in an interval by itself, and each foundational subtrace is placed in an interval by itself. Now choose any partition Δ' of L into $O(g)$ intervals whose restriction to L_B is the one just defined. From now on, when we speak of **intervals**, we will always mean those belonging to this partition of L .

Divide the edges of G into 7 subgraphs, all on the vertex set V . Let G_N^i be the graph consisting of all nonplanar edges incident to exactly i critical vertices, where $i \in \{0, 1, 2\}$. Let G_1, G_2, G_3, G_4 be the 4 pages of G_P .

Notice that G_N^2 has only $O(g)$ edges, the largest number that can possibly be induced by $O(g)$ vertices in a genus g graph (Euler's Formula). Using the equivalence relation $\sim_{\Delta'}$, we obtain well-defined quotient chain graphs having all the desired properties for G_1, \dots, G_4 , and for G_N^2 and G_N^0 (the foundational subtraces do not interleave in L_B , and L_B preserves the F -order of vertices within each foundational subtrace.)

Our only concern then, is with edges of G_N^1 . Let us sidestep for a moment.

Suppose we have p indexed subsets R_1, \dots, R_p of L with the property (#):

(#) For every $i < j$, there is no element of R_j *strictly* between two elements of R_i .

Let R be the restriction of L to $\bigcup R_i$, and define a partition of R into **segments** as follows. Label an element with a $*$ if it belongs to more than one R_i . Label each remaining element with the index of the unique R_i that contains it. To obtain the partition, let each contiguous set of similarly labeled unstarred elements determine a segment, and put each starred element in its *own* segment.

Lemma 3.5.1 There are at most $4p - 3$ segments.

Proof. By induction on p .

The lemma clearly holds for $p = 1$. So suppose it holds for p . We will show that it holds for $p + 1$. Let $\Gamma = \{R_1, R_2, \dots, R_{p+1}\}$ be an indexed set system with property (=). Then the system $\Gamma' = \{R_2, R_3, \dots, R_{p+1}\}$ also has property (=). By the induction hypothesis, Γ' determines at most $4p - 3$ segments. Since no set in Γ' has an element strictly between two elements of R_1 , the addition of R_1 to Γ' can introduce at most 4 segments (two corresponding to the smallest and largest elements of R_1 , one corresponding to the interior of R_1 , and one resulting from the possibility that R_1 splits an already existing segment of Γ'). Hence Γ determines at most $(4p - 3) + 4 = 4(p + 1) - 3$ segments. ■

In a few moments, we will use this lemma to help us understand the nature of G_N^1 . Given a trace T in F (so that $F = UTV$ for traces U and V), define the **old vertices** of T to be the set of vertices (necessarily articulation points) that appear both in T and in U . Let the **new vertices** of T be the set of vertices that occur in T but not in U .

Associated with each homotopy class C and each critical vertex v of C , is a subtrace $T_{C,v}$, consisting of all the vertices (the vertices of a "fan") to which v is

adjacent by an edge of C . Let T_1, \dots, T_p be the sequence of such subtraces ordered as they appear in F . Lemma 3.2.4 tells us $p = O(g)$. Let O_i denote the old vertices of T_i . The following useful lemma shows us that the O_i do not interleave too badly in L .

Lemma 3.5.2. The indexed set system O_1, \dots, O_p has the property (#).

Proof. Suppose for contradiction, that for some $i < j$ there is an $x \in O_j$ and $u, v \in O_i$ such that $u <_L x <_L v$. Think of the sequence F as being along a horizontal line, so early encounters are to the left and later encounters to the right. Since $u <_L x <_L v$, the primary occurrences of u , x , and v in F appear in the order u, x, v . Since $v \in O_i$, notice T_i is a segment of F whose left endpoint is on or to the right of the primary occurrence of v . Finally, since $i < j$, we have that T_j is a segment of F whose left endpoint is on or to the right of the right endpoint of T_i . Using the fact that $u \in T_i$, and $x \in T_j$, we see that F contains the subsequence u, x, v, u, x (not necessarily consecutive). But this is impossible because F cannot contain any subsequence of the form a, b, a, b where $a = b$. ■

Therefore, by Lemma 3.5.1, the induced labeling of $O = \bigcup O_i$ determines a partition of O into at most $4p - 3$ segments.

Divide the edges of G_N^1 into 2 subgraphs, G_{old} and G_{new} , each on the vertex set V . For each homotopy class C and each terminal vertex v of C , throw into G_{old} all edges that connect v to old vertices of $T_{C,v}$. Place into G_{new} all edges that connect v to new vertices of $T_{C,v}$. Figure 3.19, shows what a typical G_{old} looks like with its vertices linearly ordered according to L . The critical vertices have been elevated above the ordering to make the picture easier to understand. Each critical vertex is labeled with the associated set of old vertices it covers. (Notice that property (#) is satisfied.)

The equivalence relation $\sim_{\Delta'}$ yields a well-defined quotient chain graph for G_{new} having all the desired properties because new vertices of T_1, \dots, T_p do not interleave

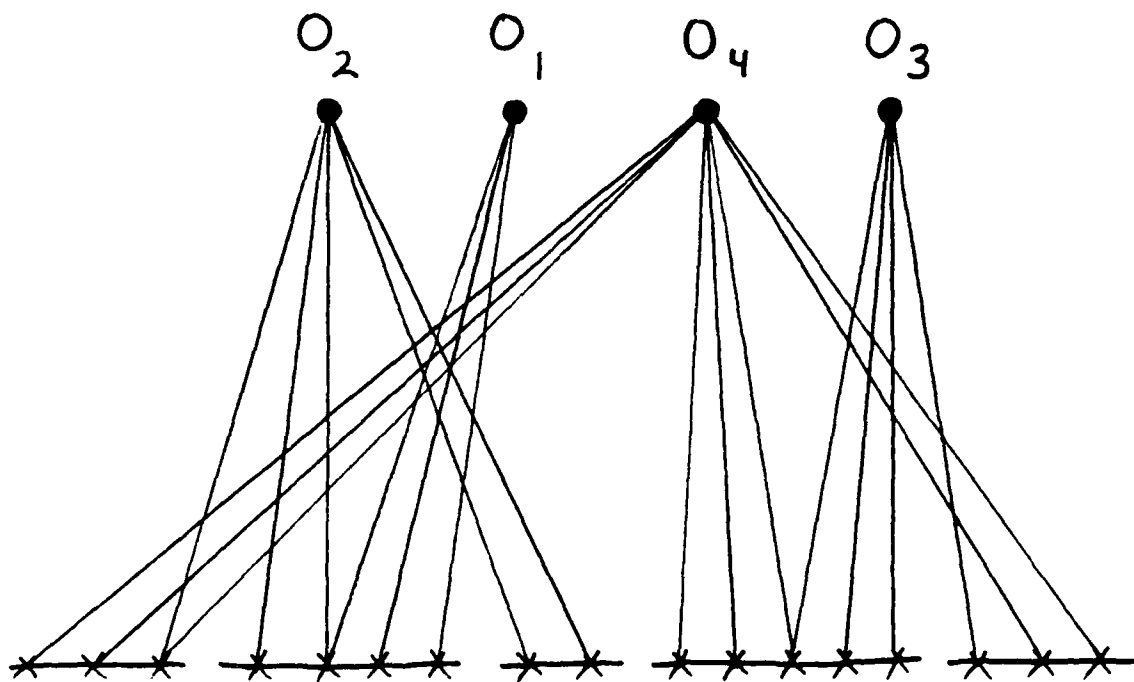


Figure 3.19: A typical $G_{[4]}$ with its vertices linearly ordered according to L .

in L .

On the other hand, old vertices of T_1, \dots, T_p may very well interleave in L . In order to get a well-defined quotient chain graph for G_{old} , we must introduce a new equivalence relation on edges. Say two edges of G_{old} are equivalent iff they connect the same pair of intervals *and they are incident to the same segment of O* .

Lemmas 3.5.1 and 3.5.2 imply that G_{old} has only $O(g)$ equivalence classes. Notice that now many equivalence classes may join a critical vertex to a single interval. In order to get a quotient chain graph H_{old} that is star-linked, construct H_{old} from G_{old} as before except this time represent critical vertices by chains with only one vertex (and potentially high degree.) Now H_{old} is star-linked, has $O(g)$ chains and $O(g)$ edges, and represents which edges of G_{old} can be placed on the same page with respect to any permutation of the intervals.

We have decomposed the graph G into the 8 subgraphs $G_1, G_2, G_3, G_4, G_N'', G_N^2, G_{\text{new}}$, and G_{old} , and obtained a quotient chain graph for each. Now to finish the proof, do a simultaneous application of the Chain Graph Lemma to all of these quotient chain graphs.

We have just established our main result:

Theorem 3.5.1. Genus g graphs have pagewidth $O(\sqrt{g})$.

3.6 A Las Vegas Algorithm

As in the edge-case, there is a Las Vegas algorithm to embed a genus g graph in $O(\sqrt{g})$ pages, although here we need the genus g surface embedding as part of the input. The algorithm works as follows. First take the input graph G and embed it in a book with $O(g)$ pages using the Heath-Istrail algorithm. (This is where we need the initial surface embedding as part of the input.) Let L be the linear ordering of the vertices so obtained. Partition L into intervals as described in the last section, and obtain the associated family of quotient chain graphs H_i . Pick a

random permutation π of the intervals and partition the edges of each H_i into levels as in the Chain Graph Lemma. Using the deterministic polynomial-time version of Lemma 3.3.1, we can embed all the j -level edges of H_i in the optimal number of pages. Doing this for each H_i and all j , gives us a book embedding of G . If this embedding uses more than $O(\sqrt{g})$ pages, then pick another random permutation of the intervals and repeat. Continue picking random permutations until a book embedding that uses $O(\sqrt{g})$ pages is found. By the proof of the Chain Graph Lemma, this algorithm runs in expected polynomial time.

Chapter 4

An Embedding for the Mesh of Cliques

4.1 Introduction

The $n \times n$ **Mesh of Cliques**, $M(n)$, is the graph whose vertex set is $\{1, 2, \dots, n\} \times \{1, 2, \dots, n\}$ and whose edges connect each row $\{i\} \times \{1, 2, \dots, n\}$ into an n -vertex clique and each column $\{1, 2, \dots, n\} \times \{i\}$ into an n -vertex clique. The Mesh of Cliques was mentioned by Chung, Leighton, and Rosenberg [CLR] as a particularly nice example of a regular graph with unknown pagenumber. They demonstrate (nonconstructively) that $M(n)$ has pagenumber $O(n^2)$, and show that any book embedding of $M(n)$ which orders the vertices row-by-row along the spine requires $n^{4/3}$ pages. (No nontrivial lower bound is known for unrestricted vertex orderings.)

We can achieve a tighter (nonconstructive) upper bound by applying Theorem 2.2.1, which gives $O(n^{3/2})$ pages. However, there is something a little unsatisfying about using a nonconstructive argument on a graph with so much structure. Certainly, we would prefer an algorithmic way to embed the graph. In addition, it would also be of interest to know if we could achieve the $O(n^{3/2})$ pages with a row-by-row ordering of the vertices.

It turns out, there is an explicit procedure for embedding the Mesh of Cliques in $O(n^{3/2})$ pages with a row-by-row ordering of the vertices. We devote the next section to describing this embedding. The method is combinatorially elegant, and contains ideas that might prove useful in obtaining a deterministic polynomial-time algorithm to embed an *arbitrary* graph in something close to $O(\sqrt{E})$ pages.

4.2 The Embedding

For technical reasons, we first embed $M(n)$ as a subgraph in $M(p^2)$, where p is a prime between $n^{1/2}$ and $2n^{1/2}$. (That such a prime exists follows from Theorem 8.6 of Niven and Zuckerman [NZ].) This embedding should map each row of $M(n)$ to a unique row of $M(p^2)$. Since the pagewidth of $M(n)$ is no bigger than that of $M(p^2)$, it suffices to show that $O(p^3)$ pages can accommodate $M(p^2)$ with a row-by-row ordering of the vertices. Our strategy for doing this is as follows.

- (1) Partition each row of $M(p^2)$ into p **supernodes** of size p so that only a constant number of edges span any pair of supernodes from *different* rows.
- (2) Linearly order the vertices of $M(p^2)$ simultaneously row by row and supernode by supernode.
- (3) Use Lemma 2.2.2 to embed all edges spanning vertices of the same row in p^2 pages, and all edges spanning vertices of different rows in $O(p^3)$ pages.

To show that step (1) is possible, we need a simple algebraic lemma.

Lemma 4.2.1. Let F be the field consisting of $\{0, 1, \dots, p-1\}$ under the usual operations $+$ and \cdot modulo p . For every $k \in \{1, 2, \dots, p\}$, the set of polynomials of degree $k-1$ over F (allowing leading coefficients to be 0) has precisely p^k elements, and no two of them agree in k or more places.

This lemma is applied as follows.

Arbitrarily index the rows of $M(p^2)$ with the 2-vectors of F^2 . Next divide the nodes of $M(p^2)$ into vertical **bands** containing p contiguous columns each. Number the bands $0, \dots, p-1$ from left to right. Number the columns within each band $0, \dots, p-1$ from left to right.

We now establish a 1-1 correspondance between the quadratic polynomials over F (allowing leading coefficients to be 0) and the supernodes of $M(p^2)$. Each supernode will have exactly one vertex in each band, and all of its vertices in the same row. We shall say the " i th vertex" of a supernode is that which is in band i . The correspondance is as follows. If $f(x) = ax^2 + bx + c$, then supernode S_f lies in row (a, b) , and the i th vertex of S_f belongs to column $f(i) \bmod p$ in band i .

Notice that when f and g differ by a constant, S_f and S_g are disjoint supernodes in the same row. By the lemma, no pair of supernodes from different rows share more than two columns. Thus we have a partition of the desired kind. ■

We have proven

Proposition 4.2.1. *There is a polynomial time procedure to embed the $n \times n$ Mesh of Cliques in $O(n^{3/2})$ pages with a row-by-row ordering of the vertices.*

4.3 Comments

There is a similar strategy that can be applied to an arbitrary graph, although it is largely nonconstructive. We can show that for an arbitrary graph, there exists a partition of the nodes into $\sqrt{E/\log E}$ supernodes of roughly equal size, such that the largest matching spanning any fixed pair of supernodes is of size $O(\log E)$. This observation led us to an initial $O(\sqrt{E \log E})$ upper bound on pagewidth for arbitrary E -edge graphs.

An interesting and natural open question suggested by this approach is the following. Does there exist a partition of the nodes of an arbitrary E -edge graph into \sqrt{E} supernodes of roughly equal size, such that the largest matching spanning

any fixed pair of supernodes is of constant size?

Chapter 5

Two Lower Bounds for Regular Graphs

5.1 Introduction

Section 5.2 demonstrates that most d -regular graphs on n nodes require $\Omega(\sqrt{d}n^{1/2-1/d})$ pages, which is tight for $d \geq \log n$. This substantially improves the $\Omega(n^{1/2-1/d} \log^2 n)$ lower bound of Chung, Leighton, and Rosenberg [CLR]. The crux of the argument is a lemma which shows that most d -regular graphs are nearly “complete” when suitably large clusters of vertices are identified. We suspect the lemma may have application to a variety of other graph embedding problems. In Section 5.3, for instance, it is used to derive an area lower bound for a multilayer grid model introduced by Aggarwal, Klawe, and Shor [AgKS].

5.2 Pagenumber Lower Bound for the Class of d -regular Graphs

Say that a graph is (α, p) -dense if for every partition of the nodes into p equal supernodes, there are at least $\alpha \binom{p}{2}$ distinct supernode pairs that are spanned by an edge.

Lemma 5.2.1. Most n -vertex d -regular graphs are $(\Omega(p^{-2/d}), p)$ -dense for $p = O(\sqrt{dn})$.

Proof. For convenience we shall let $E = dn/2$.

To start, we want to get a good estimate on the number of vertex-labeled d -regular n -node graphs. The number of such graphs is certainly larger than the number of d -regular bipartite graphs whose LEFT nodes are labeled $\{0, 1, \dots, n/2 - 1\}$, whose RIGHT nodes are labeled $\{n/2, n/2 + 1, \dots, n - 1\}$, and which satisfy the property that LEFT vertices

$$\{1 + j(n/2d), 2 + j(n/2d), \dots, (j+1)(n/2d)\}$$

cover all RIGHT vertices, for every $j \in \{0, \dots, d-1\}$. The number of these graphs is precisely

$$\binom{n/2}{d, d, \dots, d}^d = \frac{(n/2)!^d}{(d!)^{n/2}} = [\Theta(n/d)]^{dn/2} = [\Theta(n/d)]^E.$$

Next we want to estimate the number of partitions of n labeled nodes into p equal labeled supernodes. This is less than the number of partitions into p labeled supernodes without regard to size, which is precisely p^n .

Given n labeled vertices and a partition of the vertices into p equal supernodes, the number of d -regular graphs that have edges spanning at most $\alpha \binom{p}{2}$ distinct supernode pairs, is certainly less than the number of E -edge graphs satisfying the

same property. The number of such E -edge graphs is bounded above by

$$\binom{\binom{p}{2}}{\alpha \binom{p}{2}} \binom{p \binom{n/p}{2} + \alpha \binom{p}{2} (n/p)^2}{E} < 2^{p^2} \binom{\Theta(\alpha n^2)}{E} = \frac{[\Theta(\alpha n^2)]^E}{E!} = [\Theta(\alpha n/d)]^E.$$

Hence the number of vertex-labeled n -node d -regular graphs whose edges span at most $\alpha \binom{p}{2}$ distinct supernode pairs for some partition of the nodes into p equal labeled supernodes, is less than $p^n [\Theta(\alpha n/d)]^E$.

We seek the largest α for which this quantity smaller than the total number of d -regular graphs. That is, we want the maximum α subject to $p^n [\Theta(\alpha n/d)]^E < [\Theta(n/d)]^E$. Simplifying, we have $p^n \alpha^E < [\Theta(1)]^E$. Hence the largest such α is $\Omega(p^{-n/E})$. ■

Now we use this lemma to prove

Theorem 5.2.1. Most n -vertex d -regular graphs require $\Omega(\sqrt[n]{dn^{1-2-1/d}})$ pages.

Proof. By Lemma 5.2.1, most d -regular n -node graphs G are $(\Omega(n^{-1/d}), \sqrt[n]{dn})$ -dense. Here we are using the fact that $(dn)^{-1/d} = \Omega(n^{-1/d})$. Fix a linear ordering of G 's vertices and partition the ordering into $\sqrt[n]{dn}$ intervals of size $n / \sqrt[n]{dn}$. Obtain a linearly ordered quotient graph $Q(G)$ by collapsing, in place, each interval to a single vertex and identifying multiple edges. $Q(G)$ has $\Omega(n^{-1/d}(dn))$ edges since G is $(\Omega(n^{-1/d}), \sqrt[n]{dn})$ -dense. In any book embedding of $Q(G)$, less than $2\sqrt[n]{dn}$ edges can appear on the same page (an outerplanar graph on t nodes has less than $2t$ edges.) Thus $\Omega(n^{-1/d}\sqrt[n]{dn})$ pages will be required to embed $Q(G)$, and hence $\Omega(n^{-1/d}\sqrt[n]{dn})$ pages will be required to embed G . ■

5.3 An Area Lower Bound for Multilayer VLSI

We suspect that Lemma 5.2.1 or results like it will have application to a variety of graph embedding problems. This section furnishes one such application. Here

we derive an area lower bound for a multilayer VLSI grid model introduced by Aggarwal, Klawe, and Shor [AgKS]. Following [AgKS], the k -PCB (PCB stands for printed circuit board) model consists of k grid layers. Each node of the embedded graph is placed in the same position on each layer, and edges are embedded as paths in the grid which may change layers at contact cuts. An edge path may begin and end on any layer, but within each layer, paths cannot intersect except at endpoints. [AgKS] develops several algorithms all of which focus on the case where contact cuts are not allowed. This is a very desirable property since cuts may require expensive fabrication techniques and more area, and tend to make a chip less reliable.

The **thickness** of a graph G is the minimum number of planar subgraphs into which it can be decomposed (the planar subgraphs are allowed to share vertices but not edges.) Clearly, a graph can be embedded in the k -PCB model without contact cuts iff it has thickness at most k .

Many of the results in [AgKS] deal with a variant of the k -PCB model in which each node of the embedded graph must be placed in a specified location. This is referred to as the **fixed placement model**. This model arises naturally in the design of printed circuit boards where often the placement of the nodes must respect certain constraints.

Proceeding more formally, a **placement** is a one-to-one mapping of the nodes of a graph G into a set of disjoint horizontal segments of a rectangular grid R . We say that an embedding of G in R **respects a placement** σ if the nodes are embedded according to σ . A **universal placement** is a set N of n horizontal segments in a rectangular grid R with the property that for some class C of n -node graphs, if G is any graph in C and σ is any placement of the nodes of G onto N , then there is an embedding of G in R which respects σ . Say that an embedding algorithm for a class of graphs **respects fixed placements** if there is a universal placement for that class, such that for any graph G in the class and any placement σ onto the universal placement set, the embedding algorithm gives an embedding of G which

respects σ .

Combining the results of [AgKS] with the lower bound in [AKLLW], and the algorithm in [KS], demonstrates that for $k = 1$ or 2 , the area required by optimal k -PCB embedding algorithms (using no contact cuts) for n -node graphs of thickness k is $\Theta(n^3)$ if the algorithms respect fixed placements, and only $\Theta(n^2)$ area otherwise.

In particular, the following result of [AgKS] is a simple consequence of the fact that bounded-degree expander graphs have quadratic crossing number (see [Lti]).

Theorem. There are n -node graphs of degree at most 3 (and therefore thickness at most 2) such that for any fixed k , every k -PCB embedding requires $\Omega(n^2)$ area, regardless of the number of contact cuts.

In the other direction, [AgKS] establishes the following

Theorem. There is an $O(n^3)$ area k -PCB embedding algorithm for n -node graphs of thickness at most k , which uses no contact cuts and respects fixed placements.

These results suggest a natural open question posed by Aggarwal, Klawe, and Shor: close the gap between the $\Omega(n^2)$ lower bound and $O(n^3)$ upper bound for the area required to embed a graph of thickness k in k layers (without contact cuts) when $k \geq 3$.

Although we cannot say anything new with regard to arbitrary placements, we can say something interesting if we assume the nodes of G are mapped to a single horizontal line in the grid. We call such a mapping a **linear placement**.

To prove the next proposition, we need the notion of **crossing number** of a graph. The crossing number of a graph G (denoted $\nu[G]$) is the minimum number of edge-crossings achieved by some embedding of G in the plane.

Proposition 5.3.1. For $d \geq 6$, there is a d -regular graph (therefore thickness at most $d/2$) on n nodes that requires $\Omega(n^{2.5-3/d} \sqrt{d})$ area for any $(d/2)$ -PCB embedding that uses no contact cuts and respects linear placement.

Proof. By Lemma 5.2.1, most d -regular n -node graphs are $(\Omega(n^{-1/d}), \sqrt{dn})$ -dense. It is easy to show that most d -regular graphs have bisection width $\Omega(dn)$. Hence there is some d -regular graph G that satisfies both properties.

Suppose we have some $d/2$ -PCB embedding, ϕ , that places the nodes of G along a horizontal line L and uses no contact cuts. Since the bisection width of G is $\Omega(dn)$, the height of the layout is $\Omega(dn)$. To show that the width of the layout is large, we must prove that edges in some layer cross the line L many times.

Partition L into \sqrt{dn} segments, each containing $\sqrt{n/d}$ nodes of G . Collapse the nodes of each segment to a single point and identify multiple edges. This yields a quotient graph $Q(G)$ with $\Omega(dn^{1-1/d})$ edges by Lemma 5.2.1. Hence there must be a subgraph H of G consisting of $\Omega(n^{1-1/d})$ edges, no two of which get identified in $Q(G)$, and all residing on the same layer, say layer 1.

A theorem due to Leighton [Lt2] says that any graph with E edges on t nodes, where $E \geq 4t$, has crossing number $\Omega(E^2/t^2)$. Applying this to $Q(H)$, shows that its crossing number is $\Omega(n^{2-3/d-d})$.

Let P be the collection of paths along L which connect consecutive nodes of each segment. (Thus P is the union of \sqrt{dn} disjoint paths of length $\sqrt{n/d}$.) Take H' to be the union of H and P . Since the maximum degree of any node in $Q(H)$ is at most \sqrt{dn} , it is not difficult to see that $\nu(H') \geq \nu(Q(H)) \cdot \sqrt{dn} = \Omega(n^{3/2-3/d-d^{3/2}})$. Now think of H' embedded in layer 1, so that the edges of H are mapped according to ϕ , and edges of P are straight-line segments along L . Since the edges of H do not cross each other, all crossings in H' must result from edges in H crossing edges in P . Hence the line L is crossed $\Omega(n^{3/2-3/d-d^{3/2}})$ times by the edges of H , which means the width of the layout is $\Omega(n^{3/2-3/d-d^{3/2}})$. ■

Another interesting variant of the question posed by Aggarwal, Klawe, and Shor, assumes that each edge is assigned a specified layer beforehand (insisting that the edges assigned to any given layer induce a planar graph.) Even in this more restricted version of the question, we are not able to say anything new regarding

arbitrary placements. We have no results for linear placements that supercede Proposition 5.3.1.

Chapter 6

Conclusion and Open Questions

6.1 Summary of the Results

We have shown that the class of E -edge graphs has pagewidth (\sqrt{E}) , and that the class of genus g graphs has pagewidth $O(\sqrt{g})$. Both results are tight and substantially improve previously known bounds. In addition, we provided Las Vegas algorithms to generate book embeddings that satisfy these upper bounds. (The Las Vegas algorithm that embeds a genus g graph in $O(\sqrt{g})$ pages required that the initial genus g surface embedding be provided as part of the input.) All of these results could have application to some interesting problems in VLSI design.

In the other direction, we demonstrated that most d -regular graphs on n nodes require $\Omega(\sqrt{dn^{1/(2-1/d)})}$ pages, which is tight for $d > \log n$. This is also a marked improvement over previous results.

We established an area lower bound for networks embedded in multilayer grids. We proved there are n -node graphs of thickness $k \geq 3$ that require $\Omega(n^{2.5-3/2k} \sqrt{k})$ area in the k -PCB model of [AgKS] when the vertices are linearly arranged. This narrows the gap between the previous bounds of $\Omega(n^2)$ and $O(n^3)$ area.

Finally, we described a constructive $O(n^{3/2})$ -page embedding for the $n \times n$ Mesh of Cliques graph that orders the vertices row-by-row along the spine. This nearly

meets the $n^{4/3}$ lower bound for such an ordering established in [CLR].

6.2 Open Problems

The following is a list of open problems that concern book embeddings and related issues addressed in this thesis.

1. Is there a deterministic polynomial-time algorithm to embed an E -edge graph in $O(\sqrt{E})$ pages? Is there a polynomial time algorithm to embed a d -regular bipartite graph in $O(\sqrt{dn})$ pages, where all left vertices precede all right vertices? (Such an algorithm is known for $d = 2$, but not for any larger d .)
2. Is the pagenumber of nonorientable genus g graphs $O(\sqrt{g})$? (See Section 3.2 for the definition of nonorientable genus.)
3. Section 2.3 gives a deterministic polynomial-time algorithm to embed an arbitrary n -node graph in $\log n$ times optimal pages with respect to a specified ordering of the vertices. Is there a deterministic polynomial-time constant-factor approximation algorithm for this problem? A graph G is a **circle graph** if there is a 1-1 correspondance between the nodes of G and chords of a circle such that two nodes of G are adjacent iff the corresponding chords intersect. Section 2.3 shows that the chromatic number of an n -node circle graph is at most $\log n$ times the size of the largest clique. Is it the case that the chromatic number is at most a constant times the size of the largest clique? What can we say about the chromatic number if there are no triangles?
4. Given a graph G and a drawing ϕ of G in the plane, let $I(G, \phi)$ be the associated **intersection graph**, defined as follows. Each node of $I(G, \phi)$ corresponds to an edge of G , and each edge (e_1, e_2) of $I(G, \phi)$ indicates that the edges e_1 and e_2 of G cross in the drawing ϕ . Define $\omega(G)$ to be the

minimum over all ϕ , of the size of the largest clique that appears in $I(G, \phi)$. Is $\omega(G)$ equal to the thickness of G , $\theta(G)$? Does equality hold if $\theta(G) = 3$?

5. What is the pagewidth of the $n \times n$ Mesh of Cliques? What is the pagewidth of the $n \times k$ Mesh of Cliques when we restrict ourselves to a row-by-row ordering of the vertices along the spine? A sharp answer to the latter question for all k would be a nice generalization of Lemma 2.4 and Theorem 2.3 in [CLR].
6. Does there exist a partition of the nodes of an arbitrary E -edge graph into \sqrt{E} supernodes of roughly equal size, such that the largest matching spanning any fixed pair of supernodes is of constant size?
7. What is the pagewidth of the Mesh of Trees? DeBruijn Graph? Shuffle-Exchange Graph? An Expander Graph? Since these are all trivalent graphs, the algorithm of Theorem 4.10 in [CLR] can embed them all $O(\sqrt{n})$ pages.
8. Is the lower bound of Theorem 5.2.1 tight for the class of d -regular graphs when $d = \log n$? We suspect not.
9. Give an explicit construction of d -regular graphs on n nodes that require $\Omega(\sqrt{dn^{1/2-1/d}})$ pages.
10. What is the worst-case area required by a thickness k graph in the k -PCB model, with and without prespecified layer assignments for the edges? What if we assume the vertices are linearly ordered?
11. Does every n -node 3-page graph have $o(n)$ bisection width? Such a result would have important implications in complexity theory since Turing machine computation graphs are embeddable in three pages. (A paper by Galil, Kannan, and Szemerédi [GKS] suggests that the answer may be “no”).

12. Fix a graph G , and let B be a book embedding of G . The **pagewidth** of a page in B is the maximum number of edges on the page that cross any line perpendicular to the spine. Define the **width** of B to be the maximum pagewidth of any page in B . Chung, Leighton, and Rosenberg ask the following questions: (1) Is there a fixed number p such that all n -node 1-page graphs can be realized in p pages of width proportional to $\log n$? (2) Is there a fixed number q such that all n -node 2-page graphs can be realized with q stacks of width proportional to \sqrt{n} ?

Heath [He2] answered the first question affirmatively, but the second question remains open.

Index

(α, p) -dense 79

book 9

book embedding 9

canonical ordering

 of a 2BG 19

 of a 2BCG 57

chain graph 54

chain subgraph 56

circle embedding 10

completely crossing

 matching 19

 2BCG 57

critical vertices 53

crossing number 82

degenerate trace 46

endpoints of a trace 28

faces of an embedding 25

foundational subtrace 53

graph 9

homotopic edges 31

homotopy class 31

orientable 36

nonorientable 36

conclusive 46

transitional 46

linear ordering of chains 56

orientable genus

 of a graph 25

 of a surface 25

nonorientable genus

 of a graph 25

 of a surface 25

nonplanar edges 28

1-linked 56

page 9

pagenumber 10

permutation of chains 56

planar-nonplanar decomposition 28

quotient chain graph 63

quotient graph 56

rotation 27

safe edge 43

safe vertex 43

spine 9

star-linked 56

subrotation 27

trace 28

Bibliography

- [AgKS] A. Aggarwal, M. Klawe, P. Shor (1985). *Multilayer grid embeddings for VLSI*. To appear in *Algorithmica*.
- [AKLLW] A. Aggarwal, et al (1985). *A lower bound on the area of permutation layouts*. To appear in *Algorithmica*.
- [BK] F. Bernhart, B. Kainen (1979). *The book thickness of a graph*. *Journal of Combinatorial Theory B* 27, 320-331 .
- [BL] S. Bhatt, T. Leighton (1984). *A framework for solving VLSI graph layout problems*. *J. comput. Syst. Sci.*, 28, pp. 300-343.
- [BS] J. Buss, P. Shor (1984). *On the pagenumber of planar graphs*. 16th Annual ACM Symp. on Theory of Computing, 98-100.
- [CLR] F. Chung, T. Leighton, A. Rosenberg (1987). *Embedding graphs in books: a layout problem with application to VLSI design*. *SIAM Journal on Algebraic and Discrete Methods* 8, 33-58.
- [Di] R. Dilworth (1950). *A decomposition theorem for partially ordered sets*. *Ann. Math. Ser. 2* 51, 161-166.
- [Ed] J. Edmonds (1960). *A combinatorial representation for polyhedral surfaces*. *Notices of the American Mathematical Society* 7, 646.
- [EI] S. Even, A. Itai (1971). *Queues, stacks, and graphs*, *Theory of Machines and Computations*, Z. Kohavi and A. Paz eds., Academic Press, New York, pp. 71-86.
- [FMR] I. Filotti, G. Miller, J. Reif (1979). *On determining the genus of a graph in $O(v^{O(g)})$ steps*, 11th ACM Symposium on Theory of Computing, 27-37.
- [GJMP] M. Garey, D. Johnson, G. Miller, C. Papadimitriou (1980). *The complexity of coloring circular arcs and chords*. *SIAM Journal on Algebraic and Discrete Methods* 8, 216-227.

- [GKS] Z. Galil, R. Kannan, E. Szemerédi (1986), *On nontrivial separators for k -page graphs and simulations by nondeterministic one-tape Turing machines*, 18th ACM Symp. on Theory of Computing, 39–49.
- [Ha] F. Harary (1969), *Graph Theory*. Addison-Wesley Publishing Company, Reading, Massachusetts.
- [He1] L. Heath (1984), *Embedding planar graphs in seven pages*, 25th IEEE Symposium on Foundations of Computer Science, 74–83.
- [He2] L. Heath (1985), *Algorithms for embeddings graphs in books*, Dissertation, Department of Computer Science, University of North Carolina at Chapel Hill.
- [HI] L. Heath, S. Istrail (1987), *The pagenumber of genus g graphs is $O(g)$* , 19th ACM Symp. on Theory of Computing, 388–397.
- [Is] S. Istrail (1985), *An algorithm for embedding planar graphs in six pages*, Department of Mathematics, Wesleyan University, unpublished manuscript.
- [Ka] R. Kannan (1985), *Unraveling k -page graphs*, Information and Control 66, 1–5.
- [KS] M. Klawe, P. Shor (1986), *Optimal universal layouts*, Preprint.
- [Ls] C. Leiserson (1980), *Area-efficient graph layouts for VLSI*, 21st IEEE Symp. on Foundations of Computer Science, pp. 270–281.
- [Lt1] T. Leighton (1984), *New Lower Bound Techniques for VLSI*, Math Systems Theory 17, pp.47–70.
- [Lt2] T. Leighton (1981), *Layouts for the Shuffle-Exchange Graph and Lower Bound Techniques for VLSI*, Ph.D. Thesis, Dept of Math., MIT, 1981.
- [Lt3] T. Leighton (1988), personal communication.
- [LR1] T. Leighton, R. Rosenberg (1983), *Automatic generation of three-dimensional circuit layouts*, IEEE International Conference on Computer Design, pp. 633–636.
- [LR2] T. Leighton, R. Rosenberg (1986), *Three-dimensional circuit layouts*, SIAM Journal of Computing, pp. 793–813.
- [NZ] I. Niven, H. Zuckerman (1980), *An Introduction to the Theory of Numbers*, John Wiley and Sons Inc.

- [PPST] W. Paul, N. Pippenger, E. Szemerédi, W. Trotter (1983), *On determinism versus nondeterminism and related problems*, 24th IEEE Symposium on Foundations of Computer Science, 429-438.
- [Ro1] A. Rosenberg (1981), *Three-dimensional integrated circuitry*, VLSI Systems and Computations, H.T. Kung, B.Sproull, G.Steele eds., Computer Science Press, Rockville, Maryland, pp. 69-80.
- [Ro2] A. Rosenberg (1983), *Three-dimensional VLSI: a case study*, Journal of the ACM, 30, pp. 397-416.
- [Ro3] A. Rosenberg (1983), *The Diogenes approach to fault-tolerant arrays of processors*, IEEE Transactions on Computing C-32, 902-910.
- [So] H.C. So (1974), *Some theoretical results on the routing of multilayer printed wiring boards*, 1974 IEEE International Symposium on Circuits and Systems, pp. 296-303.
- [Ta¹] R. Tarjan (1972), *Sorting using networks of queues and stacks*, Journal of the ACM, 19, pp. 341-346.
- [TK] B.S. Ting, E.S. Kuh (1978), *An approach to the wiring of multilayer printed circuit boards*, 1978 IEEE International Symposium on Circuits and Systems, pp. 902-911.
- [Wh] A. White (1984), *Graphs, groups and surfaces*, Elsevier Science Publishers, Amsterdam.
- [Wi] A. Wigderson (1982), *The complexity of the hamiltonian circuit problem for planar graphs*, Princeton University EECS Department Report 298.
- [Ya] M. Yannakakis (1986), *Four pages are necessary and sufficient for planar graphs*, 18th Annual ACM Symp. on Theory of Computing.